



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Um estudo de transformações matemáticas em pontos de referência em Algoritmos de Otimização com Muitos Objetivos

Dissertação de Mestrado

Matheus Carvalho de Oliveira



São Cristóvão – Sergipe

2021

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Matheus Carvalho de Oliveira

**Um estudo de transformações matemáticas em pontos de
referência em Algoritmos de Otimização com Muitos
Objetivos**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): André Britto de Carvalho e Myriam Regattieri De Biase da Silva Delgado

São Cristóvão – Sergipe

2021

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE**

Oliveira, Matheus Carvalho de
O48e Um estudo de transformações matemáticas em pontos de
referência em algoritmos de otimização com muitos objetivos ;
orientador André Britto de Carvalho. - São Cristóvão, 2021.
70 f. : il.

Dissertação (mestrado em Ciência da Computação) –
Universidade Federal de Sergipe, 2021.

1. Computação. 2. Otimização matemática. 3. Algoritmos. I.
Carvalho, André Britto orient. II. Título.

CDU 004

Dedico esse trabalho a minha mãe, com muito amor, como forma de retribuição pelo o carinho, tempo e atenção que foram investidos em mim durante a minha criação, sem os quais seria impossível chegar onde cheguei. De certa forma, sinto até que esse trabalho também pertence a você. Somente você é capaz de saber o quanto foi difícil chegar até aqui.

De seu querido filho, Matheus.

Agradecimentos

A minha querida mãe Gicelma que se sacrificou de corpo e alma para criar seus filhos, e dar a eles tudo que ela não teve, sem pedir nada em troca.

A toda minha família, em especial aos meus irmãos Leonardo e Elias e meu pai por existirem em minha vida.

Ao meu querido professor André Britto e sua querida esposa Beatriz Trinchão, que me guiaram pelo o caminho da ciência, e me ajudaram a alcançar meu sucesso acadêmico e profissional.

Aos professores e funcionários do DCOMP/UFS e também a todos que contribuíram diretamente e indiretamente para que eu chegasse até aqui.

O homem não é nada além daquilo que a educação faz dele. (Immanuel Kant)

Resumo

Problemas de Otimização de Muitos Objetivos (MaOPs, do inglês *Many Objective Optimization Problems*) são problemas que possuem mais de três funções objetivo a serem otimizadas. A maioria dos Algoritmos Evolutivos Multiobjetivo possui baixa escalabilidade quando o número de funções objetivo aumenta. Para enfrentar essa limitação, novas estratégias têm sido propostas. Uma delas é o uso de pontos de referência para aprimorar a capacidade de busca dos algoritmos. O NSGA-III é um algoritmo baseado em pontos de referência e vem sendo aplicado com sucesso para resolver os MaOPs. Ele usa um conjunto de pontos de referência localizados em um hiperplano normalizado o qual é igualmente inclinado a todos os eixos dos objetivos e intercepta cada eixo de forma equidistante no nível 1, 0. Apesar dos bons resultados do NSGA-III, a forma da hiper-superfície que serve de referência não é explorada de forma extensiva na literatura. Este trabalho procura propor um algoritmo capaz de explorar a relação entre os pontos de referência e a melhoria da busca em um contexto de otimização de muitos objetivos. Num primeiro momento são propostas três transformações do conjunto de pontos de referência utilizados pelo NSGA-III. Além disso, o procedimento de Adaptação Guiada por Vetor (RVEA) também é aplicado para modificar periodicamente o conjunto de pontos de referência do NSGA-III original. Num segundo momento do desenvolvimento da proposta, um novo algoritmo (K-Greedy) é apresentado, tendo como principal característica realizar as transformações nos pontos de referência de maneira autônoma, a partir de um conjunto de transformações disponíveis. Nos experimentos, avalia-se o desempenho das transformações propostas tanto de forma isolada (na primeira etapa) quanto integradas no conjunto de transformações possíveis realizadas pelo algoritmo K-Greedy (na segunda etapa do desenvolvimento). Nestes experimentos, as versões original e adaptadas do NSGA-III são confrontadas considerando-se vários problemas de *benchmarking*, observando a convergência e a diversidade através da análise de testes estatísticos. Os resultados mostram que as transformações, em especial aquelas realizadas de forma alternada pelo K-Greedy, são capazes de prover melhorias no NSGA-III sem deteriorar o desempenho à medida que se incrementa o número de objetivos.

Palavras-chave: Otimização com Muitos Objetivos, Pontos de Referência, NSGA-III, Algoritmo Guloso.

Abstract

Many-Objective Optimization Problems (MaOPs) are problems that have more than three objective functions to be optimized. Most Multi-Objective Evolutionary Algorithms scale poorly when the number of objective functions increases. To face this limitation, new strategies have been proposed. One of them is the use of reference points to enhance the search of the algorithms. NSGA-III is a reference point based algorithm that has been successfully applied to solve MaOPs. It uses a set of reference points placed on a normalized hyper-plane which is equally inclined to all objective axes and intercepts at 1.0 each axis. Despite the good results of NSGA-III, the shape of the hyper-surface that supports the search is not deeply explored in the literature. This work seeks to propose an algorithm capable of exploring the relation between reference points and the improvement of the search in the optimization of many objective problems. At first, we propose three different mechanisms to transform the set of reference points used by NSGA-III. In addition, the Vector Guided Adaptation (RVEA) procedure is applied to modify periodically the original NSGA-III set of reference points. In a second stage of the development, a new algorithm (K-Greedy) is presented, whose main characteristic is to perform the transformations of the reference points autonomously from a set of available transformations. In the experiments, the performance of the proposed transformations is evaluated both, in separated way (in the first stage) and also when integrated in a pool of the K-Greedy algorithm (in the second stage). In these experiments, the original and adapted versions of the NSGA-III are confronted considering several problems of benchmarking, observing the convergence and diversity through the analysis of statistical tests. The results show that the transformations, especially those alternately carried out by K-Greedy, are able to provide improvements in the NSGA-III without deteriorating the performance when the number of objectives increases.

Keywords: Many-Objective Optimization, Reference Points, NSGA-III, Greedy Algorithm.

Lista de ilustrações

Figura 1 – Fluxograma do algoritmo NSGA-III.	21
Figura 2 – Adaptação do ponto de referência. Fonte: (Ibrahim et al., 2017)	23
Figura 3 – Demonstração da projeção da fronteira de Pareto real cobrindo apenas partes do hiperplano. Fonte: (Zhao et al., 2018)	24
Figura 4 – Processo de redistribuição dos pontos. As imagens estão em ordem cronológica, desta forma as imagens mais a direita estão em estados mais avançados do método. Fonte: (Zhao et al., 2018)	25
Figura 5 – Algoritmos mais utilizados na área.	28
Figura 6 – Ranking dos MOEA/D mais utilizados	29
Figura 7 – Principais indicadores de qualidade	29
Figura 8 – Principais testes estatísticos	30
Figura 9 – Principais Problemas de Benchmark	30
Figura 10 – Pontos de referência gerados com $p=20$	33
Figura 11 – Transformação <i>Convexa</i>	34
Figura 12 – Transformação <i>Côncava</i>	34
Figura 13 – Transformação <i>Senoidal</i>	34
Figura 14 – Representação dos pontos de referência côncavo e convexos em um mesmo plano tridimensional.	39
Figura 15 – As soluções não dominadas obtidas por cada algoritmo no DTLZ2 com 3 objetivos na melhor execução associada ao valor do hipervolume normalizado.	42
Figura 16 – As soluções não dominadas obtidas por cada algoritmo no CDTLZ2 com 3 objetivos na melhor execução associada ao valor do hipervolume normalizado.	43
Figura 17 – 1-Greedy-MaOEA: conjunto R^{new} obtido após a aplicação de até uma única transformação matemática	47
Figura 18 – 2-Greedy-MaOEA: conjunto R^{new} obtido após a aplicação de até duas transformações matemáticas	47
Figura 19 – Evolução da distribuição dos pontos para o Algoritmo 1-Greedy	48
Figura 20 – Calculo do hipervolume para uma Fronteira de Pareto de 2 objetivos: o ponto nadir (em vermelho) e quatro soluções não dominadas (com uma delas em azul com sua contribuição específica para o cálculo do hipervolume).	51
Figura 21 – As coordenadas paralelas da fronteira de soluções não dominadas obtidas por cada algoritmo no DTLZ2 com 10 objetivos na melhor execução associada ao valor do hipervolume normalizado.	63
Figura 22 – As coordenadas paralelas da fronteira de soluções não dominadas obtidas por cada algoritmo no CDTLZ2 com 10 objetivos na melhor execução associada ao valor do hipervolume normalizado.	64

Lista de abreviaturas e siglas

MOEAs	Algoritmos de Otimização com Muitos Objetivos, do inglês <i>Multi-Objective Evolutionary Algorithms</i>
EA	Algoritmo Evolucionário, do inglês <i>Evolutionary Algorithm</i>
MaOPs	Problemas de Otimização com Muitos Objetivos, do inglês <i>Many-Objective Optimization Problems</i>
PF	Fronteira de Pareto, do inglês <i>Pareto Front</i>
GD	<i>Generational Distance</i>
IGD	<i>Inverted Generational Distance</i>
HV	Hipervolume, do inglês <i>Hypervolume</i>
EMO	Otimização Evolucionária com Muitos Objetivos, do inglês <i>Evolutionary Multi-Objective Optimization</i>
MOP	Problema de Otimização com Muitos Objetivos, do inglês <i>Multi-Objective Optimization Problem</i>
NSGA-III	<i>Non-dominated Sorting Genetic Algorithm-III</i>
NSGA-II	<i>Non-dominated Sorting Genetic Algorithm-II</i>

Sumário

1	Introdução	12
1.1	Objetivos	14
1.2	Metodologia da Pesquisa	15
1.3	Principais Contribuições	16
1.4	Estrutura do Trabalho	16
2	Revisão da Literatura	17
2.1	Otimização Multiobjetivo	17
2.1.1	Conceitos Básicos	17
2.1.2	Vetor de Funções Objetivo	18
2.1.2.1	Dominância de Pareto	18
2.1.3	Conjunto Pareto Ótimo e Solução Pareto ótima	18
2.1.3.1	Vetor Ideal	19
2.1.3.2	Fronteira de Pareto	19
2.2	Algoritmos Evolutivos Multiobjetivo	19
2.2.1	MOEAs baseados em Pareto e Pontos de Referência	20
2.2.1.1	NSGA-III	20
2.2.1.2	RVEA	20
2.2.2	Outras abordagens Relacionadas	22
3	Questões de Pesquisa	27
3.1	Algoritmos estado-da-arte	28
3.2	Indicadores de qualidade	29
3.3	Testes estatísticos	30
3.4	Principais (instâncias) de problemas	30
3.5	Considerações finais sobre as questões de pesquisa	30
4	Algoritmos Base: transformações isoladas do hiperplano do NSGA-III	32
4.1	Estudo da influência das transformações matemáticas	32
4.2	Transformações em pontos de referência	35
4.2.1	Indicadores de qualidade e visualização dos resultados	35
4.2.2	Algoritmos e Parâmetros	36
4.2.3	Resultados e Análises	37
4.2.4	Principais Conclusões	40
5	Algoritmo K-Greedy: transformações conjugadas do hiperplano do NSGA-III	44

5.1	Algoritmo K-Greedy MaOEA	45
5.2	Metodologia	48
5.2.1	Problemas de Benchmark	48
5.2.2	Indicador de Qualidade e Visualização dos Resultados	50
5.2.3	Algoritmos e Parâmetros	52
5.3	Resultados e Discussão	53
5.3.1	Efeito do K-Greedy sobre o NSGA-III	55
5.3.2	Análise dos parâmetros F_r e N_r	57
5.3.3	Comparação contra algoritmos do estado da arte	58
5.3.4	Análise de complexidade dos algoritmos	60
5.3.5	Comparação de Tempo de Execução	61
6	Conclusão	65
6.1	Principais Contribuições	65
6.2	Destaques	65
6.3	Trabalhos futuros	66
	Referências	67

1

Introdução

Problemas de otimização multiobjetivo são problemas para os quais existem várias medidas, em geral conflitantes, que podem ser usadas para identificar a qualidade das soluções encontradas por algoritmos de otimização ([CHANKONG; HAIMES, 1983](#)). Esse tipo de problema é bastante comum em aplicações práticas como por exemplo os carros autônomos ([MA et al., 2019](#)), quando na concepção dos diferentes modelos é preciso equilibrar conforto e preço, ou no planejamento do controle automático deve-se balancear economia de combustível, desempenho, desgaste do motor e assim por diante. Outros exemplos de problemas de otimização multiobjetivo envolvem circuitos eletrônicos ([FLEMING; PURSHOUSE; LYGOE, 2005](#)), quando se considera o tamanho, dissipação de calor e desempenho, ou então redes elétricas ([OLIVER; KIPOUROS; SAVILL, 2014](#)), quando tratam de estabilidade, desempenho, segurança, sustentabilidade. Esses são apenas alguns exemplos, mas grande parte dos problemas do mundo real pode, a depender do nível de verossimilhança considerado em suas modelagens, envolver otimização multiobjetivo.

Em um contexto de medidas ou objetivos conflitantes como os descritos anteriormente, uma solução é chamada de não dominada, se nenhuma das medidas (cada uma avaliada por uma função objetivo diferente) puder ter seu valor aprimorado sem degradar algum dos outros objetivos ([TAKAHASHI, 2011](#)). O conjunto de todas as soluções não dominadas, chamado de conjunto de soluções Pareto-ótimas ([TAKAHASHI, 2011](#)), forma no espaço dos objetivos a superfície denominada fronteira de Pareto. Neste tipo de problema, não há somente uma melhor solução, e sim um conjunto de soluções não comparáveis, devido à necessidade de se considerar diferentes níveis de balanceamento (*trade-off*) entre as medidas conflitantes. Dados os desafios impostos pelos conflitos entre os diferentes objetivos e portanto a dificuldade de se encontrar o conjunto de soluções Pareto-ótimas, há um grande interesse em explorar esse tipo de problema via algoritmos de otimização multiobjetivo, e um interesse particular pelos algoritmos evolucionários (MOEAs do inglês *Multi-Objective Evolutionary Algorithms*). Seleção baseada na dominância de Pareto é a base de muitos MOEAs bem conhecidos, como por exemplo o NSGA-II ([DEB et al., 2002a](#)), um algoritmo bastante utilizado para resolver problemas multiobjetivo.

A otimização de problemas com muitos objetivos (MaOPs, do inglês *Many Objective Optimization Problems*) refere-se à otimização de problemas com quatro ou mais objetivos e representa um contexto ainda mais desafiador para os algoritmos de otimização. Recentemente, essa área de pesquisa tem atraído cada vez mais atenção na comunidade de otimização (DEB; JAIN, 2014). O interesse por algoritmos que trabalham com um grande número de objetivos cresceu devido às limitações dos MOEAs (SCHÜTZE; LARA; COELLO, 2011). Com isso, esforços em pesquisas têm sido orientados em direção à investigação da escalabilidade desses algoritmos com respeito ao número de objetivos, ou seja, avaliar a capacidade do algoritmo para encontrar o conjunto de soluções Pareto-ótimas (ou uma boa aproximação deste conjunto) à medida que o número de objetivos aumenta (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008). A principal razão para essa propriedade de escalabilidade é que o número de soluções não dominadas aumenta exponencialmente com o número de objetivos. Li et al. (2015) comentam que o alto número de soluções não dominadas e a baixa pressão em direção à fronteira de Pareto podem comprometer o desempenho dos MOEAs e aproximar os resultados produzidos por estes algoritmos daqueles fornecidos por uma busca aleatória (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008).

Tendo em vista a dificuldade dos MOEAs tradicionais na resolução de MaOPs, novas técnicas têm sido propostas para superar estas limitações (CUATE et al., 2017), como por exemplo aquelas que propõem: modificar a tradicional definição de dominância de Pareto, combinar o critério da dominância de Pareto com outras métricas relacionadas à convergência desenvolver novos critérios de seleção baseado em indicadores de performance. Outras alternativas envolvem o uso de um conjunto de pontos de referência para guiar distribuição das soluções ao longo da busca. Baseados nesta ideia, destacam-se dois algoritmos que tiveram bons resultados lidando com MaOPs e servirão de base para as abordagens propostas neste trabalho: NSGA-III (DEB; JAIN, 2014) (*Non-dominated Sorting Genetic Algorithm III*) e RVEA (CHENG et al., 2016) (*Reference Vector Guided Evolutionary Algorithm*).

A principal contribuição do NSGA-III para resolver os MaOPs está em seus pontos de referência que são distribuídos de maneira uniforme sobre um hiperplano que corta os eixos de forma equidistante. O mecanismo de diversidade proposto fortalece os membros não dominados da população mais perto do conjunto de pontos de referência fornecidos para assegurar a diversidade. Apesar do sucesso do NSGA-III, a distribuição dos pontos de referência ainda não foi profundamente investigada na literatura e as variações, em geral, envolvem apenas o total de pontos no conjunto. O RVEA aparece com destaque uma vez que prevê distribuição não uniforme dos pontos sobre o hiperplano. Entretanto, embora o RVEA seja o primeiro algoritmo a lidar com as transformações de pontos de referência, ele apenas ajusta dinamicamente seus pontos de distribuição para melhor representar a solução. Verifica-se assim uma lacuna na literatura com a possibilidade de se investigar transformações da hiper-superfície usada de referência, por exemplo através da introdução de não-linearidade.

Ainda no contexto de alternativas para lidarem com MaOPs, aparecem os MOEAs baseados em indicadores, que podem adotar diferentes indicadores para guiar o processo evolutivo, como hipervolume, ϵ -unário, GD e IGD. Exemplos de tais algoritmos são IBEA (ZITZLER; KUNZLI, 2004) e SMS-EMOA (BEUME; NAUJOKS; EMMERICH, 2007). Neste contexto, um dos mais conhecidos é o IBEA (*Indicator-Based Selection in Multiobjective Search*). Este pode ser definido como um algoritmo evolutivo baseado em indicadores gerais pois considera diferentes tipos de indicadores de desempenho de acordo com a preferência do usuário. O SMS-EMOA é um algoritmo de estado estacionário mais relacionado com a abordagem proposta no presente trabalho, pois utiliza o indicador de hipervolume aliado ao conceito de dominância de Pareto. Outro exemplo de abordagem relacionada é encontrado em (Tian et al., 2018), onde um indicador de desempenho é usado para ajustar um conjunto de pontos de referência com base nas contribuições de soluções candidatas em um arquivo externo.

1.1 Objetivos

Este trabalho propõe uma alternativa, mais especificamente um algoritmo guloso denominado de *K-Greedy*, para a escolha, de forma dinâmica (ou seja ao longo da busca), das transformações não lineares a serem aplicadas sobre o hiperplano do NSGA-III. Esta proposta vem amparada em experimentos preliminares apresentados em (Oliveira; Britto, 2018), nos quais observou-se que a mudança de pontos de referência provou-se capaz de alterar significativamente o desempenho do NSGA-III. As mudanças nos pontos de referência foram realizadas por três transformações matemáticas diferentes, chamadas convexas, côncavas e senoidais e foram testadas em problemas *Benchmark* ao transformar o conjunto de pontos de referência em cada variante diferente do NSGA-III. No presente trabalho, diferentemente de (Tian et al., 2018) que usa o IGD como indicador de desempenho, considera-se o aumento/redução do hipervolume como base para aceitar/rejeitar as modificações realizadas no conjunto de pontos de referência como forma de melhorar o desempenho do algoritmo. Assim, a abordagem proposta neste trabalho pode, portanto, ser categorizada como uma abordagem híbrida de MaOEA que mistura ideias de MOEAs baseados em indicadores, uma vez que usa hipervolume como indicador de desempenho, com aqueles de dominância de Pareto, porque adota NSGA-III como a estrutura principal para a resolução dos MaOPs. No presente trabalho, no entanto, apesar de também considerar transformações matemáticas em pontos de referência, a abordagem proposta seleciona aleatoriamente que tipo de transformação será aplicada em quais pontos e um mecanismo baseado em indicadores decide aceitar ou rejeitar tais transformações. Os resultados obtidos pelo algoritmo guloso proposto (*K-Greedy*) foram apresentados e discutidos em (Oliveira; Delgado; Britto, 2021). As abordagens gulosas podem ser definidas como aquelas que sempre levam a melhor solução imediata ou local ao tomar uma decisão. Embora algoritmos gulosos possam encontrar soluções sub-ótimas para alguns problemas, especialmente aqueles com superfícies multimodais, eles podem encontrar rapidamente soluções aceitáveis para problemas complexos.

Ainda, de forma mais específica, pretende-se:

- Propor um algoritmo híbrido (baseado em indicador e dominância de Pareto) para resolver MaOPs;
- Explorar transformações não lineares sobre a o hiperplano do NSGA-III de maneira positiva;
- Investigar uma estratégia aleatória de escolha dos pontos de referência a serem transformados;
- Avaliar uma estratégia gulosa para tomada de decisão sobre aceitar ou não a transformações realizadas;
- Estudar a influência da modificação dos pontos de referência sobre o desempenho do NSGA-III;
- Avaliar a qualidade das técnicas propostas com base em testes estatísticos e análise das fronteiras de Pareto geradas.

1.2 Metodologia da Pesquisa

A metodologia utilizada no trabalho consiste de uma pesquisa exploratória, que busca investigar uma lacuna encontrada no mapeamento sistemático, no qual foram selecionadas técnicas de otimização com muitos objetivos que utilizam pontos de referência. Novas técnicas são então propostas com o intuito de preencher a lacuna encontrada na pesquisa exploratória. Em seguida, realiza-se um conjunto de testes em *benchmarks*, para se verificar a eficácia das técnicas propostas. Esse estudo também permite avaliar qual dos algoritmos elencados é mais adequado. As conclusões dessa primeira etapa de desenvolvimento permitem prever a possibilidade de agrupar algumas das transformações em uma galeria a ser disponibilizada para um só algoritmo realizar a escolha da transformação mais adequada ao longo do processo de otimização. Este algoritmo prevê o uso de uma estratégia gulosa para decidir se aceita ou rejeita as transformações realizadas.

Nos experimentos verifica-se a eficácia das modificações comparando os algoritmos propostos (variantes NSGA-III) com o original, utilizando testes estatísticos para avaliar o aprimoramento das técnicas propostas. Também são feitas comparações com outros algoritmos para avaliar a proposta no contexto do estado da arte. São consideradas 12 instâncias de MaOPs para avaliar o desempenho dos algoritmos comparados. Para isso, nos experimentos são consideradas sete instâncias da família WFG ([HUBAND et al., 2006a](#)) e cinco instâncias da família DTLZ ([DEB et al., 2002c](#)). Todos os algoritmos são comparados observando a convergência e a diversidade por meio da análise de testes estatísticos em cenários de múltiplos e muitos objetivos

1.3 Principais Contribuições

Este trabalho contribui com a comunidade de otimização multiobjetivo e em especial de otimização de muitos objetivos ao realizar uma análise empírica para melhor compreender a relação entre os pontos de referência e a melhoria na busca realizada por diferentes algoritmos de otimização com muitos objetivos baseados no NSGA-III. O trabalho também contribui ao propor um algoritmo que pode explorar essa relação de forma benéfica. Portanto, a novidade do trabalho é a proposta de um algoritmo autônomo que, para cada problema, seleciona a cada iteração (i) a função matemática que será aplicada e (ii) o subconjunto de pontos de referência que serão transformados. Além disso, decide aceitar ou não a transformação realizada, com base na melhora ou degradação no indicador de hipervolume.

Os resultados da pesquisa foram publicados em um trabalho de conferência e um artigo (Oliveira; Britto, 2018) de revista (Oliveira; Delgado; Britto, 2021).

1.4 Estrutura do Trabalho

Para melhor compreensão da proposta o texto está organizado nos seguintes capítulos:

- Após esse capítulo introdutório, no capítulo 2 são apresentados conceitos básicos sobre otimização multiobjetivo, com destaque para estratégias que utilizam vetor de peso e pontos de referência. O capítulo traz ainda trabalhos relacionados num contexto de otimização com muitos objetivos.
- O capítulo 3 traz questões de pesquisa que foram levantadas e discutidas durante a pesquisa sistemática realizada.
- O capítulo 4 apresenta um estudo sobre a influência dos pontos de referência, e também propõe três transformações não lineares realizadas sobre o hiperplano do NSGA-III.
- O capítulo 5 apresenta um novo algoritmo de otimização com muitos objetivos, a metodologia a ser seguida nessa dissertação, assim como o método proposto e uma breve descrição da implementação, bem como resultados.
- Por fim, no capítulo 6 apresenta-se a conclusão da dissertação e o capítulo traz ainda tópicos de trabalhos futuros.

2

Revisão da Literatura

Neste capítulo serão mostrados os conceitos básicos de otimização multiobjetivo, assim como algoritmos para a solução de problemas multiobjetivo. Também serão discutidos trabalhos relacionados existentes na literatura, tendo como foco a Otimização com Muitos Objetivos ou estratégias que utilizem pontos de referência ou vetores de peso, dando prioridade para trabalhos mais recentes.

2.1 Otimização Multiobjetivo

Otimização é um tópico muito amplo mas, de maneira bem simplificada, pode ser entendido como a tarefa de alocar recursos da melhor forma para uma certa atividade e que atenda às restrições impostas. Otimização multiobjetivo refere-se à otimização simultânea de múltiplos objetivos conflitantes. Nestes casos a solução do problema envolve não apenas uma única solução ótima, mas um conjunto de soluções (conhecidas como soluções Pareto-ótimas). Estas soluções têm a característica de serem não dominadas, isto é, todas as soluções podem ser consideradas equivalentes não sendo possível ordená-las segundo algum critério de qualidade. A seguir serão apresentados alguns conceitos que permitem um entendimento geral da formalização de tópicos relacionados à otimização multiobjetivo.

2.1.1 Conceitos Básicos

Segundo (TAKAHASHI, 2011), um vetor de variáveis de decisão $\vec{x} = (x_1, x_2, \dots, x_n)$ é o conjunto ordenado de valores numéricos pertencentes ao espaço amostral que será escolhido pelo problema (ou seja uma possível solução).

2.1.2 Vetor de Funções Objetivo

Um vetor de funções objetivo de dimensão m é definido por:

$$\overrightarrow{f(\vec{x})} = f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x}) \quad (2.1)$$

Em problemas multiobjetivo há duas ou mais funções objetivo a serem otimizadas simultaneamente. No entanto, isso nem sempre é possível pois muitos desses objetivos podem ser conflitantes. A otimização multiobjetivo é definida para um problema irrestrito de minimização por:

$$\text{Minimize } \overrightarrow{f(\vec{x})} = f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x}) \quad (2.2)$$

Quando há mais de três funções objetivo, esses problemas são chamados de Problemas de Otimização de Muitos Objetivos (MaOPs, do inglês *Many Objective Optimization Problems*).

2.1.2.1 Dominância de Pareto

Para se comparar os dois vetores de funções objetivo em um problema de otimização multiobjetivo é necessário definir a dominância de Pareto ([TAKAHASHI, 2011](#)). É dito que $\overrightarrow{f(\vec{x})}$ domina $\overrightarrow{f(\vec{x}')}$ se as condições abaixo forem satisfeitas:

1. O vetor $\overrightarrow{f(\vec{x})}$ é pelo menos igual a $\overrightarrow{f(\vec{x}')}$ em todas funções objetivo
2. O vetor $\overrightarrow{f(\vec{x})}$ é melhor que $\overrightarrow{f(\vec{x}')}$ em pelo menos um objetivo

Então para um problema de minimização a igualdade abaixo pode ser verificada:

$$\forall k \in (1, 2, \dots, m) : f_k(\vec{x}) \leq f_k(\vec{x}') \text{ e } \exists k \in (1, 2, \dots, m) : f_k(\vec{x}) < f_k(\vec{x}')$$

Um vetor de funções objetivo $f_k(\vec{x})$ que satisfaz as condições acima é dita como não-dominada ou Pareto-ótima.

O conjunto de todas soluções não dominadas forma o Conjunto Pareto-ótimo.

2.1.3 Conjunto Pareto Ótimo e Solução Pareto ótima

Um Conjunto Pareto Ótimo pode ser definido como:

$$C = \{x \in \Omega \mid x \text{ é Pareto ótimo}\} \quad (2.3)$$

Onde (Ω) é o espaço de busca da função objetivo.

2.1.3.1 Vetor Ideal

O ponto ideal $\overrightarrow{f_k(\vec{x}_0)}$ é um ponto que possui os menores valores possíveis da função objetivo $\overrightarrow{f_k(\vec{x})}$ para todos os valores de $k \in (1, 2, \dots, m)$. O vetor ideal, é o vetor que liga o ponto ideal a origem $(\vec{0})$.

2.1.3.2 Fronteira de Pareto

É formada pelos pontos no espaço das funções-objetivo que correspondem ao conjunto de soluções Pareto-ótima. O intuito de todo algoritmo de otimização multiobjetivo é o de encontrar a Fronteira de Pareto, uma vez que as soluções que lá se encontram têm a propriedade de não serem dominadas por nenhuma outra.

O objetivo dos MOEAs é mover os vetores objetivo não dominados em direção à fronteira de Pareto (convergência), e também gerar uma boa distribuição desses vetores na PF (PF, do inglês *Pareto Front*) e diversidade.

2.2 Algoritmos Evolutivos Multiobjetivo

Técnicas de otimização e particularmente aquelas baseadas em conceitos de evolução têm sido usadas para lidar com o desafio imposto por objetivos múltiplos e conflitantes durante a resolução de problemas multiobjetivo (COELLO; LAMONT; VELDHUIZEN, 2006). Como apontado em (LI et al., 2015; ZHOU et al., 2011), existem três categorias principais de algoritmos Evolutivos Multiobjetivo (MOEAs, do inglês *Multi-Objective Evolutionary Algorithms*) propostas para resolver problemas multiobjetivo: (i) Abordagens baseadas em Pareto, que adotam mecanismos baseados em dominância de Pareto para distinguir e selecionar soluções candidatas; (ii) métodos baseados em decomposição, que decompõem um problema multiobjetivo original em um número de problemas de otimização de objetivo único a serem resolvidos de forma colaborativa e (iii) algoritmos baseados em indicadores, cujos critérios de seleção estabelecem suas decisões com base em indicadores de desempenho de medidas de qualidade da solução.

Apesar dos bons resultados dos MOEAs para poucos objetivos (2 e 3), uma vez que o número de funções objetivo aumenta, os MOEAs têm dificuldades em definir a fronteira de Pareto e a busca se torna quase que completamente aleatória (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008), (SCHÜTZE; LARA; COELLO, 2011). Além da pouca escalabilidade para os MaOPs (problemas com mais de 3 objetivos), os MOEAs também sofrem com problemas com uma grande quantidade de variáveis de decisão (MALTESE; OMBUKI-BERMAN; ENGELBRECHT, 2016) (problemas de larga escala).

Para superar estas desvantagens, alguns MOEAs usam estratégias baseadas em pontos de referência, conforme detalhado na subseção a seguir.

2.2.1 MOEAs baseados em Pareto e Pontos de Referência

Essa seção descreve os dois algoritmos que servem de base para as propostas apresentadas nesse trabalho: NSGA-III e RVEA.

2.2.1.1 NSGA-III

O Non-dominated Sorting Genetic Algorithm - versão III (NSGA-III) (DEB; JAIN, 2014) proposto por Deb e Jain é um dos algoritmos mais famosos da área de Otimização com Muitos Objetivos e foi também um dos grandes propulsores da utilização da técnica dos pontos de referência no ambiente de MOEAs. Ele estende o princípio de funcionamento de seu antecessor (NSGA-II), incluindo um mecanismo que usa um conjunto de pontos posicionados em um hiperplano normalizado para orientar a pesquisa. Esses pontos de referência são utilizados principalmente para garantir a convergência e a diversidade. Isto é feito através do reforço dos membros da população não dominada que estão mais próximos a este conjunto de pontos de referência. O procedimento básico do NSGA-III é descrito na Fig. 1.

A maior contribuição do NSGA-III reside na geração do hiperplano nos passos 1, 2 e 3, onde ele define o mecanismo de diversidade baseado em pontos de referência. No fim do loop evolucionário, as melhores soluções de cada nicho são usadas para preencher as posições vagas para a próxima geração. A plataforma básica do NSGA-III é semelhante à do NSGA-II (DEB et al., 2002a). Mas ao contrário do NSGA-II a diversidade no NSGA-III é garantida fornecendo um conjunto de pontos de referência bem espalhados. O trabalho apresentado por Deb, também concluiu que o NSGA-III foi capaz de encontrar um conjunto de soluções com boa convergência e diversidade para problemas envolvendo fronteiras que apresentavam características convexas, côncavas, disjuntas, com densidade variável de pontos através da fronteira e que são escalados de forma diferente como (HUBAND et al., 2006a).

Conforme destacado anteriormente, o NSGA-III usa um conjunto de pontos de referência posicionados em um hiperplano normalizado que é igualmente inclinado a todos os eixos e tem uma intersecção em 1 em cada eixo como pode ser visto na Figura 10. Nesta figura, os pontos de referência foram escolhidos de acordo com a estrutura descrita em Das e Dennis (1998).

2.2.1.2 RVEA

Apesar do NSGA-III ter tido bons resultados com pontos de referência, ainda há muitas lacunas para serem exploradas. Desta maneira, o RVEA (CHENG et al., 2016) tenta explorar um pouco a distribuição desses pontos de referência. Os autores propõem um algoritmo que dinamicamente ajusta a distribuição dos vetores de referência com o intuito de melhorar a diversidade e convergência.

RVEA (CHENG et al., 2016) é um dos primeiros algoritmos que avaliam o efeito de modificar a distribuição de pontos de referência no desempenho de algoritmo de otimização. No

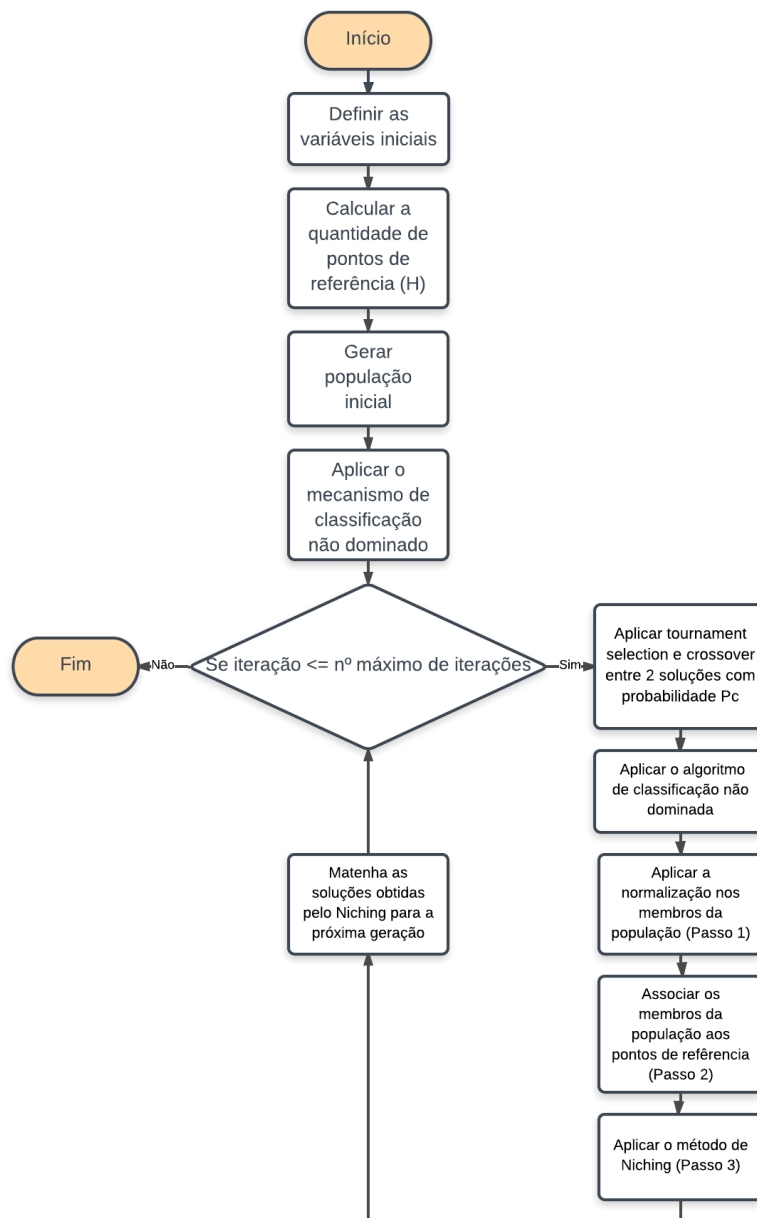


Figura 1 – Fluxograma do algoritmo NSGA-III.

RVEA, é proposta uma estratégia de adaptação para ajustar dinamicamente a distribuição dos vetores de referência de acordo com as escalas das funções objetivo. No entanto, isso só é possível se o valor da função objetivo puder ser facilmente normalizado no mesmo intervalo (CHENG et al., 2016). Infelizmente, podem existir MaOPs onde os objetivos têm diferentes faixas, como em (HUBAND et al., 2006a). Como essa estratégia visa trabalhar com objetivos escaláveis, em fronteiras geométricas irregulares não é garantido obter uma distribuição uniforme dos vetores de referência. A principal diferença entre o NSGA-III e o RVEA está na possibilidade de a RVEA adaptar seus vetores para representar melhor soluções, enquanto os pontos de referência NSGA-III são sempre distribuídos uniformemente sobre o hiperplano. Além disso, embora o RVEA seja o primeiro algoritmo proposto para lidar com as transformações dos pontos de

referência, apenas ajusta dinamicamente seus pontos de distribuição para representar melhor a solução. No entanto, nunca muda a forma da hiper-superfície.

De maneira mais técnica, o RVEA consiste de uma escalarização, denominada Abordagem Angular de Distância Penalizada e de uma estratégia de Adaptação Guiada de Vetores. No algoritmo RVEA, dado um conjunto de vetores de referência unitários e uniformemente distribuídos (como o DTLZ1), espera-se que esse algoritmo retorne um conjunto de soluções Pareto Ótimas uniformemente distribuídas, que são os pontos de intersecção entre cada vetor de referência e a Fronteira de Pareto.

2.2.2 Outras abordagens Relacionadas

Anteriormente, os algoritmos eram caracterizados pelo uso de vetores de peso ou pontos de referência normalizados e uniformemente distribuídos. A importância dos pontos de referência, já foi discutida na literatura (hiperplano duplo (DEB; JAIN, 2014), adaptação guiada de vetores (CHENG et al., 2016)). Por outro lado, a relação do formato da fronteira de Pareto com o formato dos pontos de referência, não foi investigada em nenhum estudo. Além destes trabalhos, durante os últimos 3 anos, uma grande variedade de MOEAs foi proposta na literatura (Ishibuchi; Doi; Nojima, 2016; Tian et al., 2018; Ibrahim et al., 2017; Zhao et al., 2018; Vesikar; Deb; Blank, 2018; Khan et al., 2016). A ideia básica comum nesses algoritmos é procurar soluções Pareto ótimas ao longo desses pontos, e utilizá-los para garantir diversidade e/ou convergência.

O trabalho proposto por Ishibuchi, Doi e Nojima (2016), entende que apesar dos pontos de referência terem um conceito similar aos vetores de peso, enquanto a importância dos vetores de peso tem sido enfatizada na literatura, os pontos de referência não foram bem investigados. Para isso é proposta uma adaptação do MOEA/D utilizando como benchmark alguns problemas famosos como os família DTLZ 1-4, bem como a versão com muitos objetivos do problema da mochila. O seu método de adaptação consiste em implementar uma hipereurística, que altera a posição do ponto de referência do seu algoritmo, como pode ser visto na Fig.2, onde Z_{New}^* se trata do ponto de referência adaptado, e Z^* do ponto de referência antigo e A e B são duas soluções. A Figura 2 também deixa claro o objetivo do algoritmo, pois considerando que B parece melhor do que A no que diz respeito à convergência (por estar mais próximo da PF em roxo), A é avaliada como sendo melhor que B caso o algoritmo tivesse utilizado Z^* como ponto de referência. Essa alteração somente é importante nos estágios iniciais do problema, e a medida que o número de iterações aumenta, o seu ponto de referência adaptado se aproxima do ponto de referência real. Para avaliar a qualidade de suas soluções, é adotado como métrica o Hipervolume. Apesar dos bons resultados, o algoritmo ainda depende muito de parametrização e para concluir de fato a qualidade de suas soluções ainda é necessário comparar com outros MOEAs nas mesmas condições de teste em mais problemas de Benchmark.

O trabalho apresentado por Tian et al. (2018) capitaliza com sucesso o fato de que a fronteira de Pareto afeta fortemente o desempenho do MOEA e propõe um MOEA que usa IGD-

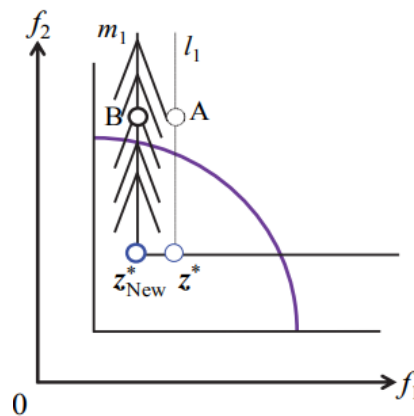


Figura 2 – Adaptação do ponto de referência. Fonte: (Ibrahim et al., 2017)

NS.¹ A modificação nos pontos de referência consiste apenas em ativar esses pontos de referência nas proximidades das soluções (dessa forma minimizando o IGD-NS). Essa metodologia obteve excelentes resultados quando comparada com: MOEA/DD, NSGA-III, RVEA e MOMBI-II. Outro resultado muito interessante é que essa estratégia pode capturar muito bem o formato e a distribuição das soluções na fronteira de Pareto. O trabalho apresentado é comparado com vários algoritmos nos mais diferentes tipos de testes de benchmark e mostra bons resultados.

Abordagens baseadas em decomposição têm recebido atenção crescente da comunidade de otimização evolutiva nos últimos anos, particularmente o algoritmo denominado otimização multiobjetivo baseada na decomposição (MOEA/D) (ZHANG; LI, 2007) devido ao seu bom desempenho na otimização numérica. A decomposição é uma estratégia básica na otimização multiobjetivo tradicional, e o MOEA/D é um dos algoritmos mais famosos desse tipo de abordagem. Ele decompõe um problema de otimização multiobjetivo em vários subproblemas de otimização escalar e os otimiza simultaneamente. Cada subproblema é otimizado usando apenas informações de seus vários subproblemas vizinhos, o que faz com que o MOEA/D tenha menor complexidade computacional a cada geração. Foi demonstrado, no artigo original, que o MOEA/D usando a normalização dos objetivos pode lidar com objetivos de escalas diferente como os da família WFG e DTLZ, e o MOEA/D com um método de decomposição avançado pode gerar um conjunto de soluções distribuídas de maneira muito uniforme para instâncias de teste com 3 objetivos.

Também utilizando hipereurísticas, Ibrahim et al. (2017) propuseram um abordagem bem diferente dos outros artigos estudados. A hipereurística proposta executa vários algoritmos conhecidos da área de otimização com muitos objetivos em paralelo, entre eles estão: GDE3, SPEA2, SMPSO e NSGA-III. Durante o processo de otimização, a abordagem proposta combina

¹ O IGD-NS é uma versão simplificada do IGD, onde ao invés de ser tomado em conta a contribuição de todas as distâncias entre solução e ponto de referência, somente as soluções mais próximas de um ponto de referência são levadas em conta. Desta forma é possível ter soluções que não são levadas em conta durante o cálculo do IGD-NS, coisa que não é possível no IGD.

as soluções de todos os algoritmos e extrai os pontos mais bem distribuídos utilizando a técnica de *non-dominated sort* e posteriormente o mecanismo de seleção do NSGA-III. A abordagem também seleciona, de maneira adaptativa, qual o melhor algoritmo para atacar o problema em diferentes pontos do processo de busca, para fazer isso ele somente conta a contribuição de soluções na população final. Desta forma, a proposta funciona em dois passos, em um primeiro passo ele utiliza vários algoritmos em paralelo e seleciona aquele que apresentou soluções mais bem distribuídas, posteriormente o melhor algoritmo é selecionado para fazer a busca de maneira individual durante algumas iterações, e depois de algum tempo o algoritmo volta a utilizar todos os algoritmos em paralelo e o *loop* se repete. A métrica utilizada para comparação dos algoritmos foi o IGD. Apesar dos bons resultados nos testes propostos, somente são realizados testes em 4 algoritmos da família DTLZ e que possuem fronteiras de Pareto regulares.

Algumas outras abordagens utilizam técnicas de *machine learning* como o SVM utilizado por Zhao et al. (2018) para periodicamente redistribuir os vetores de referência e garantir a diversidade da população. Esse tipo de estratégia é especialmente interessante pois, se bem sucedida consegue garantir a diversidade da população sem realizar cálculos extras utilizando a função de *fitness*. O algoritmo proposto chama-se fastCar e consiste de dois passos, sendo eles clusterização e redistribuição. Na etapa de clusterização é utilizado um mecanismo onde somente as linhas de referência que interceptam a Fronteira de Pareto real são utilizadas para guiar a população e garantir diversidade. A Fig. 3 ilustra o processo de clusterização.

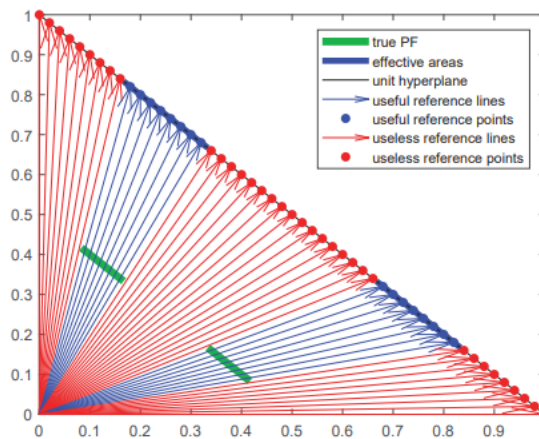


Figura 3 – Demonstração da projeção da fronteira de Pareto real cobrindo apenas partes do hiperplano. Fonte: (Zhao et al., 2018)

Geralmente os pontos de referência são uniformemente distribuídos, como os vistos na Fig. 10, e também interceptam cada objetivo em 1. Neste algoritmo, a única coisa que é mantida é a inclinação do plano, as bordas e intersecções são adaptadas automaticamente através do SVM. Uma imagem cronológica do resultado dessa etapa pode ser vista na Fig. 4. A métrica de comparação utilizada foi o IGD e NHV². E foram utilizados 15 problemas de *benchmark* da

² hipervolume normalizado entre [0, 1]

competição do CEC de 2018. O algoritmo proposto apesar de conseguir resultados competitivos, não conseguiu oferecer nenhuma vantagem que possa ser considerada disruptiva.

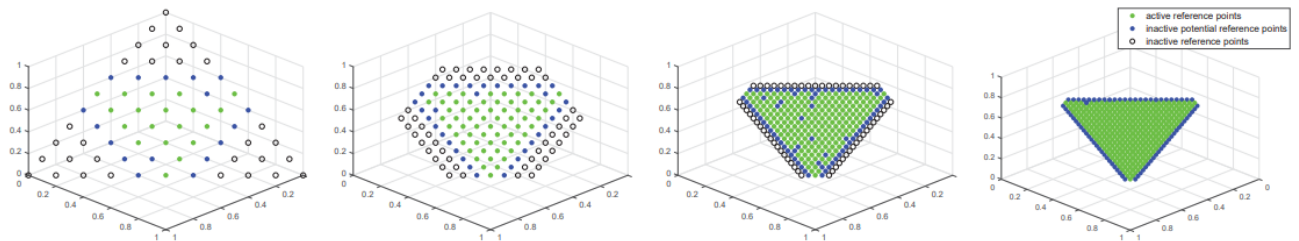


Figura 4 – Processo de redistribuição dos pontos. As imagens estão em ordem cronológica, desta forma as imagens mais a direita estão em estados mais avançados do método. Fonte: (Zhao et al., 2018)

Outros trabalhos como (Vesikar; Deb; Blank, 2018) focam na busca em uma região específica ao invés de procurar por toda a Fronteira de Pareto. Isso pode ser interessante caso o usuário tenha ideia do tipo de solução que é mais interessante para o seu problema. Esse tipo de ideia já foi sugerida anteriormente por Deb em seu trabalho original (DEB; JAIN, 2014) e também no trabalho (Oliveira; Britto, 2018) que considera a versão *Sinusoidal*, onde as soluções do problema ficaram concentradas onde haviam mais pontos de referência.

Algumas propostas como em Zhang et al. (2017) usam um arquivo para melhorar o desempenho de um algoritmo evolutivo multiobjetivo, propondo uma abordagem de arquivamento baseada em decomposição (DAA) eficiente que cria subespaços de acordo com um conjunto de vetores de peso como é MOEA/D e RVEA. Em cada geração, a abordagem escolhe apenas uma solução ótima para atualizar o arquivo em relação à diversidade. Após esta atualização, um método baseado em distância normalizado (semelhante ao NSGA-III Niching) decide onde a nova solução deve ser posicionada, e se esta solução deve substituir uma existente neste subespaço ou não. Segundo os autores, o algoritmo proposto é mais eficiente do que uma série de métodos de arquivamento de última geração em termos da diversidade de soluções não dominadas obtidas; e o DAA é capaz de acelerar a velocidade de convergência da busca evolutiva para a maioria dos problemas testados.

Na proposta de Gong et al. (2020), um framework lida com problemas de otimização multiobjetivo de intervalo dinâmico, que são muito comuns em aplicações do mundo real (como um robô que se afasta/se aproxima de uma zona de perigo). No framework, uma estratégia para decompor variáveis de decisão em dois grupos é proposta. Depois disso, duas novas subpopulações são utilizadas para otimizar cooperativamente as variáveis de decisão nos dois grupos. Além disso, uma estratégia baseada na intensidade da mudança está associada a uma estratégia de mutação aleatória para encontrar a fronteira de Pareto.

Algumas abordagens focam na seleção de recursos, como em (ZHANG et al., 2019) que propõe uma abordagem de seleção de recursos multiobjetivo englobando três novos operadores. A principal novidade consiste em um operador de mutação baseado em uma diferença de probabilidade que pode orientar os indivíduos a localizar rapidamente áreas potencialmente ótimas. O artigo também propõe um novo operador purificador de busca (OPS) para melhorar a capacidade de autoaprendizagem de indivíduos de elite e um operador de classificação não dominado usando o conceito de *crowding distance* NSGA-II (DEB et al., 2002b) para reduzir o consumo de tempo.

Alguns outros MOEAs também obtiveram bons resultados em relação aos arquivos, com destaque para a proposta apresentada em (Wang; Jiao; Yao, 2015). Esse trabalho tem como objetivo projetar um algoritmo equilibrado em relação a diferentes métricas, incluindo convergência e diversidade, porque a maioria dos algoritmos se concentra apenas em um desses aspectos. O algoritmo separa soluções não dominadas em dois arquivos (por isso é chamado de algoritmo de dois arquivos). Durante o processo de otimização, um arquivo foca na convergência e outro na diversidade. Os resultados experimentais mostram que Two Arch2 pode lidar com ManyOPs (até 20 objetivos) com convergência e diversidade satisfatórias.

Outras propostas incluem alterar o sistema de escolha de soluções não dominadas. O NSGA-III utiliza Pareto-dominância, no entanto, à medida que o número de objetivos aumenta, esse sistema perde o poder de seleção, uma vez que a grande maioria das soluções com um grande número de dimensões se torna não dominada. Para resolver esse problema, (Khan et al., 2016) propôs uma adaptação do NSGA-III com o método MOEA/D, que usa pontos de referência para transformar um problema de otimização multiobjetivo em vários problemas menores de objetivo único. Lembrando que o método de decomposição MOEA/D (ZHANG; LI, 2007) divide um MaOP em vários problemas escalares, enquanto MOEA/D-DRA (ZHANG; LIU; LI, 2009) além de dividir o problema da mesma maneira que o MOEA/D, gasta o somente poder computacional apenas nos subproblemas mais promissores. Ao fim da divisão, cada membro da população se torna um problema mono-objetivo a ser resolvido.

3

Questões de Pesquisa

As abordagens apresentadas no capítulo anterior representam as mais relevantes para o trabalho proposto encontradas na literatura. No entanto, com o intuito de nortear quais ferramentas deveriam ser usadas e quais algoritmos a serem comparados com a proposta deste trabalho, também foi feita uma pesquisa bibliográfica visando responder algumas questões importantes relacionadas à otimização multiobjetivo e de muitos objetivos:

- Quais são os algoritmos do estado-da-arte?
- Qual é medida de qualidade mais utilizada?
- Qual o método de comparação (teste estatístico) mais adotado?
- Quais as funções de *benchmark* (instâncias de problemas) mais consideradas?

Buscando responder essas perguntas, foi feita a leitura de todos os artigos relacionados nos jornais GECCO, CEC e IEEE Transaction on Evolutionary Computation, que são os jornais e conferências mais prestigiados e maiores fontes de consulta da área. Para fazer essa pesquisa, foram escolhidos os anos de 2018, 2019 e 2020. Os filtros utilizados foram: "Many-objective optimization" e "NSGA-III". Foram encontrados diversos algoritmos, no entanto somente foram investigados aqueles mais diretamente relacionado ao tema dessa pesquisa. No CEC foram encontrados 23 artigos relacionados, no GECCO foram encontrados outros 24 artigos relacionados e por fim no IEEE Transaction on Evolutionary Computation foram encontrados mais 27 artigos. Todos esses artigos foram analisados e catalogados com o intuito de responder às perguntas acima.

3.1 Algoritmos estado-da-arte

Considerando-se primeiro a popularidade de cada algoritmo investigado pode-se verificar, na Figura 5, que o algoritmo mais utilizado é o MOEA/D, seguido de NSGA-II, NSGA-III e RVEA, que são todos, com exceção do MOEA/DD, algoritmos que vão ser extremamente discutidos ao longo deste trabalho.

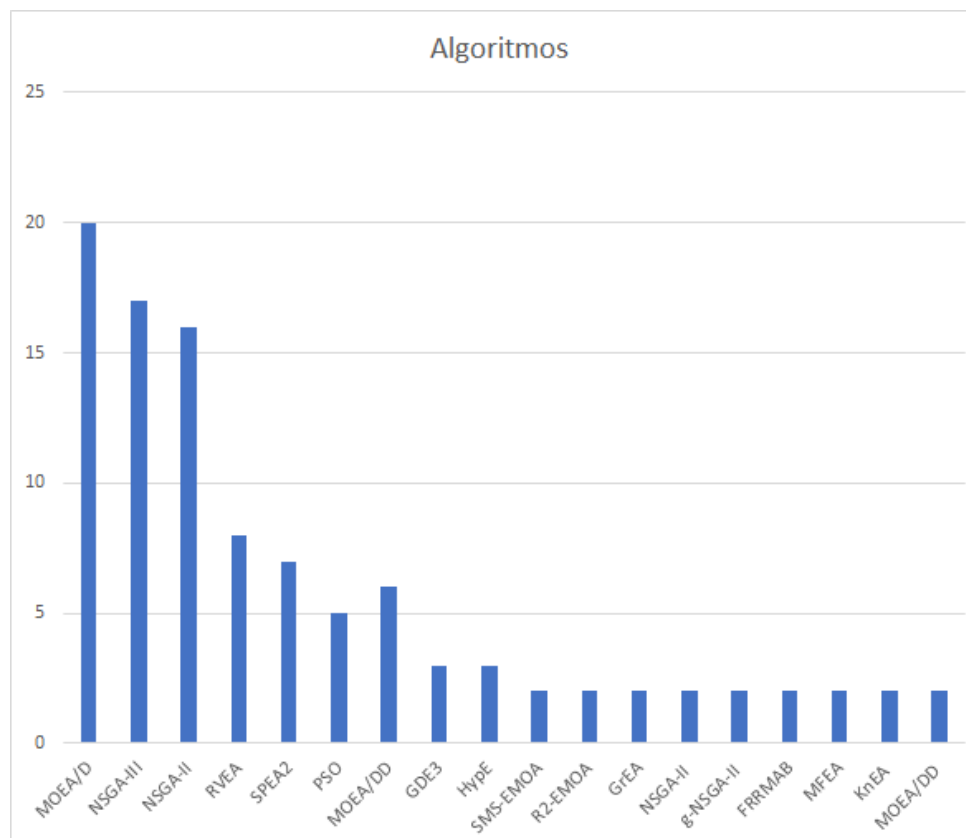


Figura 5 – Algoritmos mais utilizados na área.

Apesar do MOEA/D ser o algoritmo mais utilizado, nota-se que é um algoritmo com muitas variantes - talvez o algoritmo de otimização multiobjetivo com mais variantes que existe. A Figura 6 mostra que depois do MOEA/D original, as variantes DD e o DRA são as mais utilizadas.

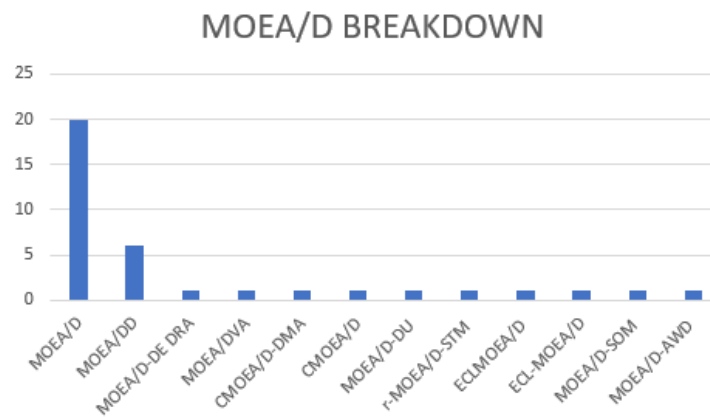


Figura 6 – Ranking dos MOEA/D mais utilizados

3.2 Indicadores de qualidade

Em termos de indicadores de qualidade, foi encontrada uma grande variedade como pode ser visto na Figura 7. Entretanto, apesar da grande quantidade de indicadores diferentes, os mais utilizados são o Hipervolume e o IGD.

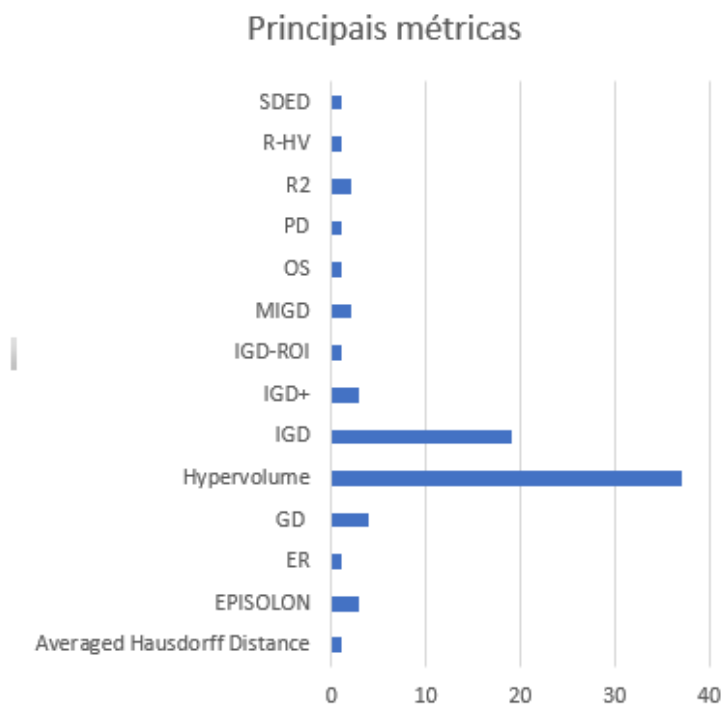


Figura 7 – Principais indicadores de qualidade

3.3 Testes estatísticos

Quando se observam os testes estatísticos, é possível notar que, exceto o Wilcoxon que está presente na maioria dos trabalhos, há certa homogeneidade para os demais, como pode ser visto na Figura 8.

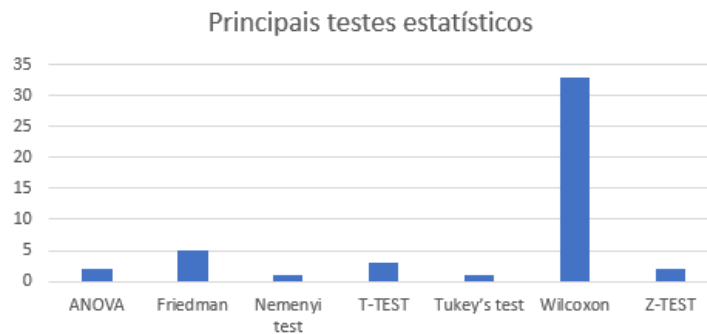


Figura 8 – Principais testes estatísticos

3.4 Principais (instâncias) de problemas

Ao se observarem os principais problemas (ou instâncias de problemas) com muitos objetivos, verifica-se uma grande diversidade. Por isso decidiu-se por somente mostrar os 5 primeiros, tendo em vista que muitos problemas eram muito específicos, e só faziam sentido dentro do trabalho daquele nicho de pesquisa, como por exemplo: aerodinâmica, mecânica. A Figura 9 mostra os principais Problemas de Benchmark.

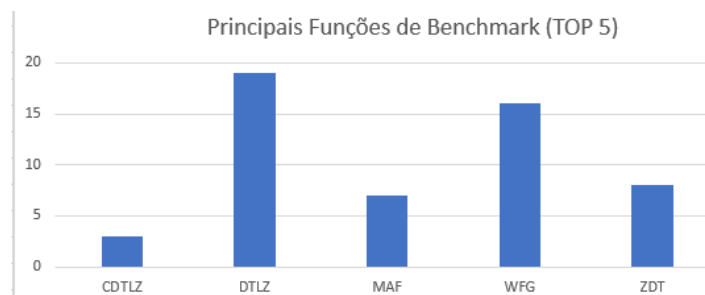


Figura 9 – Principais Problemas de Benchmark

3.5 Considerações finais sobre as questões de pesquisa

A análise realizada para responder às questões de pesquisa levantadas no início do capítulo permite concluir que a otimização de muitos objetivos tem chamado a atenção da comunidade de otimização e em especial de pesquisadores voltados para os métodos evolutivos. Constatou-se que há várias iniciativas que utilizam pontos de referência para garantir a diversidade e convergência de MOEAs à medida que o número de objetivos aumenta. O NSGA-III ganha destaque como

um dos algoritmos mais importantes neste contexto. Outra observação diz respeito à proposta com resultados mais relevantes encontrada até o momento: aquela apresentada por [Tian et al. \(2018\)](#), a qual busca manter o formato da fronteira de Pareto e somente ativar os pontos da Fronteira que tiverem mais soluções na redondeza. Apesar de vários trabalhos atestarem a importância dos pontos de referência, nenhum deles realizou modificações no formato, pois o foco das modificações foi mais voltado ao número, distribuição e bordas. A proposta deste trabalho, descrita nos capítulo 3 (primeira etapa do desenvolvimento) e que contempla o estudo de transformações aplicadas de forma isolada e posteriormente no capítulo 4 (segunda etapa do desenvolvimento) e que visa trazer sua parcela de contribuição ao abordar mecanismos autônomos de transformação não linear do hiperplano do NSGA-III, ou seja um método que adapta o formato dos pontos de referência, de maneira dinâmica, para os mais diversos MaOPs.

4

Algoritmos Base: transformações isoladas do hiperplano do NSGA-III

O NSGA-III (DEB; JAIN, 2014), é um algoritmo de otimização multiobjetivo particularmente usado em MaOPs. Conforme discutido no capítulo anterior, a principal contribuição do NSGA-III reside em seu mecanismo baseado em referência. Os pontos de referência representam, portanto, uma importante característica do algoritmo. Deste modo, uma maneira de possivelmente melhorar o desempenho do NSGA-III está em escolher automaticamente um conjunto de pontos de referência adequados para cada problema específico. Assim, estudando como diferentes conjuntos de pontos de referência podem influenciar o desempenho do NSGA-III em diferentes contextos e propor um procedimento automático para modificar seu hiperplano de acordo pode ser bastante importante, não apenas para melhorar o desempenho de MOEAs e MaOEAs existentes mas também para alavancar o design de novas abordagens para resolver os MaOPs.

4.1 Estudo da influência das transformações matemáticas

Com base nos bons resultados das abordagens baseadas em pontos de referência, aqui são propostos novos métodos de transformação e adaptação NSGA-III. O procedimento de adaptação do NSGA-III é realizado após a geração do hiperplano, onde é aplicado um procedimento de transformação. O NSGA-III gera o hiperplano original, então, um procedimento de transformação modifica a forma do conjunto de pontos de referência. Uma vez que a forma do hiperplano é alterada, ela influencia a forma como os pontos de referência são associados.

Nossa principal contribuição está nos procedimentos de transformação usados para alterar a forma do hiperplano NSGA-III. As três abordagens propostas são chamadas de: Transformação Convexa, Transformação Côncava e Transformação Senoidal. Considerando os parâmetros básicos para gerar o hiperplano discutido anteriormente, H é o número de pontos de referência, p número de divisões e m o número de objetivos, as três abordagens podem ser definidas como:

Transformação Convexa: a ideia básica é distender o hiperplano original em uma forma

convexa. O algoritmo 1 apresenta os passos de transformação, onde V_t representa o conjunto original de pontos de referência e V_{t+1} o transformado. É baseado na transformação do problema DTLZ2 para o problema CDTLZ2 apresentado em (DEB; JAIN, 2014). A Figura 11 em seus pontos azuis, mostra uma representação 3D do hiperplano transformado. Observe que ele gera uma densidade muito maior de pontos próximos à origem do espaço de coordenadas e possui uma alta densidade de pontos de referência perto de 0.

Algorithm 1 Transformação Convexa

INPUT: Pontos de referência do NSGA-III $V_t = [v_{t,0}, v_{t,1}, \dots, v_{t,H-1}]$ e número de gerações t

OUTPUT: Pontos de referência transformados $V'_{t+1} = [v'_{t,0}, v'_{t,1}, \dots, v'_{t,H-1}]$

```

1: for  $i \in \{0, \dots, H-1\}$  do
2:   for  $j \in \{0, \dots, M-1\}$  do
3:      $V_{t+1}(i, j) = (V_t(i, j))^4$ 
4:   end for
5:    $V_{t+1}(i, M-1) = (V_t(i, M-1))^2$ 
6: end for
  
```

Transformação Côncava: semelhante à transformação anterior, a ideia básica aqui é esticar o hiperplano original em uma forma côncava. O algoritmo 2 apresenta as etapas principais, seguindo as diretrizes discutidas anteriormente. A Figura 12 mostra em seus pontos azuis o resultado dessa transformação. Observe que ele tem uma forma esférica. Esta distribuição desigual de pontos de referência pode ser corrigida com a Adaptação Guiada por Vetores, uma vez que pode mover esses pontos para melhor representar a população.

Algorithm 2 Transformação Concava

INPUT: Pontos de referência do NSGA-III $V_t = [v_{t,0}, v_{t,1}, \dots, v_{t,H-1}]$ e número de gerações t

OUTPUT: Pontos de referência transformados $V'_{t+1} = [v'_{t,0}, v'_{t,1}, \dots, v'_{t,H-1}]$

```

1: for  $i \in \{0, \dots, H-1\}$  do
2:   for  $j \in \{0, \dots, M-1\}$  do
3:      $V_{t+1}(i, j) = (V_t(i, j))^{1/4}$ 
4:   end for
5:    $V_{t+1}(i, M-1) = (V_t(i, M-1))^{1/2}$ 
6: end for
  
```

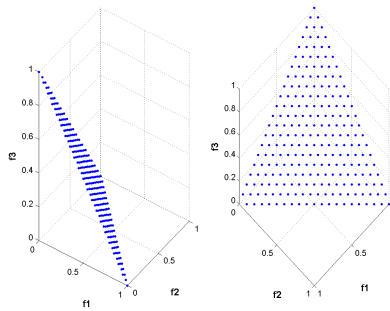


Figura 10 – Pontos de referência gerados com $p=20$.

Transformação Senoidal: aqui a transformação deforma o hiperplano original de uma forma senoidal. Esta transformação gera um conjunto de pontos de referência com algumas

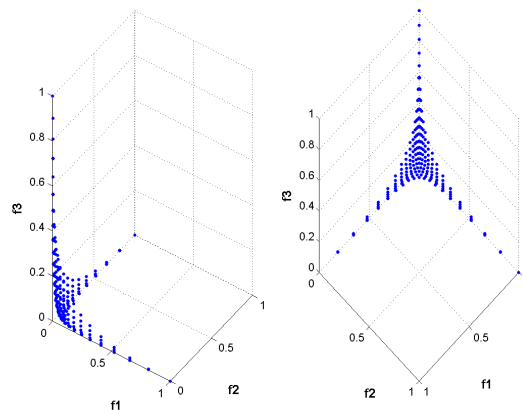


Figura 11 – Transformação *Convexa*

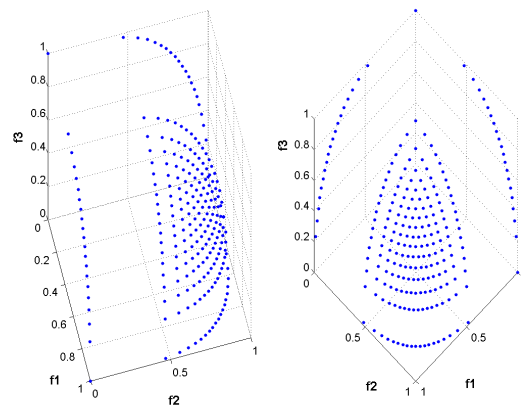


Figura 12 – Transformação *Côncava*

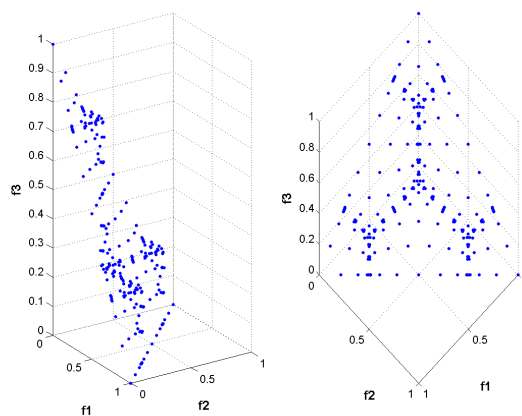


Figura 13 – Transformação *Senoidal*

propriedades da transformação côncava e convexa como pode ser visto na Fig. 13. Possui colinas positivas e negativas (considerando o hiperplano original como base). As colinas positivas concentram seus pontos nas extremidades da fronteira de Pareto, enquanto as negativas estão mais próximas da origem. Isso se deve a característica função senoidal, que começa positiva devido a frequência utilizada, e só há espaço para haver um ponto negativo em todo hiperplano.

Algorithm 3 Transformação Senoidal

INPUT: Pontos de referência do NSGA-III $V_t = [v_{t,0}, v_{t,1}, \dots, v_{t,H-1}]$ e número de gerações t

OUTPUT: Pontos de referência transformados $V'_{t+1} = [v'_{t+1,0}, v'_{t+1,1}, \dots, v'_{t+1,H-1}]$

```

1: for  $i \in \{0, \dots, H-1\}$  do
2:   for  $j \in \{0, \dots, M-1\}$  do
3:      $V_{t+1}(i, j) = (V_t(i, j)) + 0.1 \sin(4\pi V_t(i, j))$ 
4:   end for
5: end for
  
```

Além dessas três novas abordagens propostas, a transformação proposta em (CHENG et al., 2016) também é aplicada no algoritmo NSGA-III adaptado. Pode ser definido como:

$$V_{t+1,i} = \frac{v_{0,i} \circ (z_{t+1}^{max} - z_{t+1}^{min})}{\|z_{t+1}^{max} - z_{t+1}^{min}\|} \quad (4.1)$$

onde $i = 0, \dots, H-1$, v_{t+1} denota o i -ésimo vetor de referência adaptado para a próxima $t+1$ geração, $v_{0,i}$ denota o i -ésimo vetor de referência uniformemente distribuído, z_{t+1}^{max} e z_{t+1}^{min} denotam os valores máximo e mínimo de cada objetivo em $t+1$ geração, respectivamente. O operador \circ denota o produto Hadamard que multiplica dois vetores do mesmo tamanho. Este procedimento pode ser aplicado a qualquer conjunto de pontos de referência. Assim, ele pode ser combinado aos três métodos propostos anteriormente, gerando diferentes versões da transformação RVEA.

4.2 Transformações em pontos de referência

4.2.1 Indicadores de qualidade e visualização dos resultados

Na literatura, diferentes indicadores de qualidade são usados para avaliar MOEAs. Para investigar o comportamento da abordagem proposta em relação à diversidade e convergência a fronteira de Pareto, e o efeito da atual estratégia de ponto de referência utilizada, o hipervolume foi adotado. O hipervolume é um dos indicadores de qualidade mais amplamente utilizados na Otimização Multiobjetivo. Ele calcula o volume de todas as faixas retangulares até um determinado ponto nadir. Para os pontos de referência propostos posteriormente neste trabalho, o hipervolume é um indicador de maximização, portanto, quanto maior o valor de hipervolume de um algoritmo, melhor a convergência e a diversidade desta resposta. Para calcular o hipervolume é necessário selecionar um ponto nadir. Neste trabalho, $y^* = [1.5, 1.5, \dots, 1.5]$ foi usado para

DTLZ1; $y^* = [2, 2, \dots, 2]$ foi usado para DTLZ2 e DTLZ4; $y^* = [11, 11, \dots, 11]$ foi usado para DTLZ3 e $y^* = [3, 5, \dots, 2M + 1]$ foi usado para WFG1 a WFG7. Todos os valores de hipervolume apresentados neste trabalho foram normalizados entre $[0, 1]$ dividindo $\prod_{i=1}^M y_i^*$, a mesma metodologia adotada por (CHENG et al., 2016).

Foi realizado um total de 30 execuções para cada algoritmo. Para realizar uma comparação justa, para cada problema, cada algoritmo foi limitado ao mesmo número de avaliações de função objetivo. Como estávamos interessados em estudar a influência da nova transformação, usando como base o NSGA-III, foi aplicado o teste de Wilcoxon. O teste de Wilcoxon a um nível de significância 5% foi adotado para comparar esses valores de hipervolume. O teste de Wilcoxon indica se um conjunto de dados fornecido tem alguma diferença significativa entre cada conjunto de dados analisados. Em seguida, os valores médios foram usados para decidir qual algoritmo tem o melhor valor. O teste de Wilcoxon está disponível no programa estatístico R (R Core Team, 2020). Nesta comparação, cada versão adaptada do NSGA-III foi confrontada com o algoritmo NSGA-III tradicional.

Além da análise de hipervolume, foi realizada uma discussão sobre a visualização da forma da fronteira de Pareto gerada. Em nossa análise, apenas CDTLZ2 foi escolhido. Foi gerada a visualização para 3 e 10 objetivos. Na análise de 3 objetivos foi plotada a fronteira de Pareto gerada em um espaço tridimensional, para cada algoritmo. Para analisar objetivos de 10 foi utilizada a estratégia de coordenadas paralelas (LI; ZHEN; YAO, 2017). Para traçar este gráfico, selecionamos as soluções geradas pela execução com melhor hiper-volume.

4.2.2 Algoritmos e Parâmetros

O estudo discutido neste artigo analisa a influência de quatro métodos de transformação sobre os pontos de referência do NSGA-III. Três métodos são propostos aqui, côncavo, convexo e senoidal, e também é utilizado o procedimento de transformação RVEA. Além disso, cada transformação foi confrontada com NSGA-III.

Aqui, diferentes configurações de NSGA-III adaptado foram executadas, trocando apenas o método de transformação. Nosso experimento foi definido usando um total de 5 versões do NSGA-III: Côncavo (NSGA-III com Transformação Côncava); Convexo (NSGA-III com Transformação Convexa); Senoidal (NSGA-III com Transformação Senoidal); R-NSGA-III (NSGA-III com Transformação RVEA); e, NSGA-III original sem transformações. Visto que este trabalho é baseado principalmente em (DEB; JAIN, 2014) e (CHENG et al., 2016), são usados os mesmos conjuntos de parâmetros adotados nos artigos originais.

No crossover, foi utilizado o SBX Crossover com índice de distribuição de crossover definido em 30 e probabilidade de crossover definido em 1.0. Para o operador de mutação, foi utilizado uma mutação polinomial com índice de distribuição de mutação e probabilidade de mutação definida em $\eta_m = 20$ e $p_m = 1/n$, respectivamente. A condição de termino é definida

pelo número máximo de gerações. Para DTLZ1, DTLZ3 e WFG1 a WFG7, o número máximo de gerações é definido como 1000. Para DTLZ4 e DTLZ2 é definido como 500. Para 3 funções objetivo, o número de pontos de referência escolhidos (H) foi $H = 91$; para 8 funções, $H = 120$, e para 10 funções, $H = 220$. No NSGA-III, o tamanho da população é definido pelo número de pontos de referência. Assim, todos os algoritmos foram com o mesmo número de avaliações de função objetivo.

Portanto, todos os algoritmos são executados com o mesmo número de gerações. A Tabela 1 mostra o número total de pontos de referência (H) usados pelo NSGA-III, a abordagem proposta e suas variantes para diferentes números de objetivos.

Tabela 1 – Número de Pontos de Referência

No. de Objetivos	(H) Pontos de Referência
3	91
8	120
10	220

4.2.3 Resultados e Análises

O algoritmo K-Greedy foi baseado, como visto em capítulos anteriores, no trabalho publicado em (Oliveira; Britto, 2018). Desta forma, antes de entrar em detalhes a respeito do K-Greedy é discutido o resultado dos algoritmos que deram origem a ele. Isso é importante pois, os outros algoritmos são, basicamente, composição dos algoritmos bases.

A Tabela 2 apresenta os resultados do hipervolume para todos os MaOPs analisados em 3 diferentes números de funções objetivo, o valor dentro do parênteses se trata do desvio padrão, enquanto o de fora é a média aritmética de todas as execuções de cara algoritmo para cada problema.. Cada célula representa o valor médio do hipervolume para 30 execuções para um número específico de funções objetivo. Uma vez que cada algoritmo foi confrontado com o NSGA-III básico, os símbolos \approx , $+$ e $-$ indicam se a transformação foi, respectivamente, significativamente igual, melhor ou pior do que o NSGA- III. Uma comparação foi considerada sem diferença significante se $p - valor > 0,05$, no caso de haver diferença significativa (ou seja $p - valor \leq 0,05$) o algoritmo é considerado melhor se sua média for maior; e é considerado pior se sua média for menor.

Em geral, é possível influenciar a busca do algoritmo escolhendo os pontos de referência e melhorar os resultados do NSGA-III. Em linhas gerais, o NSGA-III original sustentou os melhores resultados em quase todas as comparações. No entanto, cada transformação proposta melhorou os resultados do NSGA-III em pelo menos um problema. Portanto, os pontos de referência são uma forma promissora de melhorar os resultados do NSGA-III. O R-NSGA-III, embora em alguns casos tenha melhorado os resultados do NSGA-III, não foi significativamente melhor em nenhum caso. Isso não significa que essa transformação seja ruim, significa que nestes conjunto de problemas não houve uma melhora significativa nos resultados do NSGA-III.

Tabela 2 – Resultados do Hipervolume. Comparação Principal Wilcoxon Rank Sum Test

Problema	Objetivos	NSGA-III	Côncavo	Convexo	Senoidal	R-NSGA-III
CDTLZ2	3	0.993 (0.000)	0.993 (0.000)≈	0.992 (0.000)–	0.990 (0.000)–	0.993 (0.000)≈
	8	1.000 (0.000)	0.855 (0.069)–	1.000 (0.000)≈	0.9999 (0.000)≈	1.000 (0.000)≈
	10	1.000 (0.000)	0.887 (0.078)–	1.000 (0.000)≈	0.9999 (0.000)≈	1.000 (0.000)≈
DTLZ1	3	0.992 (0.000)	0.802 (0.043)–	0.990 (0.000)≈	0.991 (0.000)≈	0.992 (0.000)≈
	8	0.986 (0.034)	0.821 (0.123)–	0.988 (0.044)≈	0.994 (0.009)≈	0.982 (0.060)≈
	10	0.999 (0.001)	0.922 (0.042)–	0.998 (0.005)≈	0.993 (0.027)≈	0.995 (0.020)≈
DTLZ2	3	0.927 (0.000)	0.701 (0.031)–	0.919 (0.000)–	0.925 (0.000)–	0.927 (0.000)≈
	8	0.986 (0.014)	0.678 (0.047)–	0.988 (0.015)≈	0.992 (0.013)≈	0.982 (0.024)≈
	10	0.986 (0.011)	0.700 (0.050)–	0.989 (0.011)≈	0.987 (0.011)≈	0.989 (0.010)≈
DTLZ3	3	1.000 (0.000)	0.951 (0.014)–	0.999 (0.001)≈	1.000 (0.000)≈	1.000 (0.000)≈
	8	0.991 (0.032)	0.876 (0.122)–	1.000 (0.001)≈	0.996 (0.008)≈	0.998 (0.005)≈
	10	1.000 (0.000)	0.943 (0.073)–	1.000 (0.000)≈	1.000 (0.000)≈	1.000 (0.000)≈
DTLZ4	3	0.859 (0.133)	0.834 (0.037)–	0.877 (0.059)+	0.852 (0.103)–	0.851 (0.064)–
	8	0.997 (0.002)	0.857 (0.071)–	0.998 (0.002)≈	0.998 (0.002)≈	0.997 (0.003)≈
	10	1.000 (0.000)	0.879 (0.043)–	1.000 (0.000)≈	1.000 (0.000)≈	1.000 (0.000)≈
WFG1	3	0.816 (0.027)	0.716 (0.044)–	0.855 (0.032)+	0.818 (0.025)≈	0.811 (0.023)≈
	8	0.703 (0.020)	0.730 (0.027)+	0.715 (0.030)≈	0.714 (0.033)≈	0.699 (0.030)≈
	10	0.709 (0.026)	0.775 (0.025)+	0.713 (0.027)≈	0.724 (0.027)≈	0.713 (0.024)≈
WFG2	3	0.906 (0.073)	0.583 (0.149)–	0.868 (0.073)–	0.898 (0.070)–	0.865 (0.078)–
	8	0.995 (0.004)	0.933 (0.077)–	0.999 (0.001)+	0.997 (0.003)≈	0.996 (0.004)≈
	10	0.999 (0.002)	0.981 (0.016)–	0.999 (0.000)≈	0.999 (0.001)≈	0.999 (0.001)≈
WFG3	3	0.711 (0.003)	0.475 (0.175)–	0.709 (0.004)–	0.694 (0.007)–	0.708 (0.003)–
	8	0.619 (0.016)	0.624 (0.024)≈	0.645 (0.015)+	0.634 (0.019)+	0.620 (0.018)≈
	10	0.602 (0.030)	0.612 (0.059)+	0.651 (0.014)+	0.634 (0.018)+	0.600 (0.025)≈
WFG4	3	0.730 (0.001)	0.397 (0.077)–	0.707 (0.002)–	0.716 (0.000)–	0.730 (0.000)≈
	8	0.929 (0.041)	0.546 (0.103)–	0.905 (0.030)–	0.900 (0.050)–	0.906 (0.067)≈
	10	0.897 (0.087)	0.642 (0.095)–	0.938 (0.036)≈	0.916 (0.076)≈	0.926 (0.047)≈
WFG5	3	0.694 (0.003)	0.399 (0.073)–	0.672 (0.002)–	0.680 (0.003)–	0.694 (0.003)≈
	8	0.895 (0.000)	0.738 (0.038)–	0.861 (0.000)–	0.881 (0.000)–	0.895 (0.000)≈
	10	0.919 (0.000)	0.779 (0.048)–	0.902 (0.000)–	0.913 (0.000)–	0.919 (0.000)≈
WFG6	3	0.685 (0.012)	0.344 (0.060)–	0.660 (0.015)–	0.666 (0.009)–	0.683 (0.015)≈
	8	0.869 (0.016)	0.453 (0.100)–	0.833 (0.016)–	0.860 (0.020)–	0.871 (0.016)≈
	10	0.886 (0.035)	0.532 (0.099)–	0.891 (0.017)≈	0.885 (0.022)≈	0.890 (0.030)≈
WFG7	3	0.731 (0.000)	0.292 (0.050)–	0.705 (0.001)–	0.717 (0.000)–	0.731 (0.000)≈
	8	0.936 (0.037)	0.474 (0.122)–	0.905 (0.027)–	0.920 (0.034)–	0.944 (0.001)≈
	10	0.953 (0.031)	0.588 (0.086)–	0.936 (0.031)–	0.937 (0.031)–	0.934 (0.033)≈

Observando os problemas do DTLZ, todos os algoritmos propostos, exceto os de formato côncavo, foram capazes de obter resultados tão bons quanto o NSGA-III. Embora DTLZ2 a DTLZ4 sejam MaOPs de forma côncava, a Transformação Convexa funcionou bem. A principal razão é que um conjunto de Pontos de Referência em formato convexo tem uma alta densidade de Pontos de Referência próximos da solução ótima 0. Portanto, atrai respostas para mais perto de uma região com um retorno maior. Em problemas DTLZ1 a DTLZ4, o melhor algoritmo permanece basicamente o mesmo, para todas as configurações. Por outro lado, o WFG1 a WFG7 mostra que o melhor algoritmo pode mudar indicando uma alta influência dos pontos de referência. Além disso, a Transformação Côncava é principalmente a pior em quase todos os algoritmos, isso reforça o efeito dos pontos de referência.

No entanto, um alto valor de hipervolume não significa necessariamente que a solução encontrada seja melhor do que outra que tenha um hipervolume um pouco menor. Em MaOPs, o usuário pode não estar interessado em encontrar toda a fronteira de Pareto. O usuário geralmente está interessado em encontrar uma parte específica da frente. Desta forma, o usuário pode saber qual é sua região preferida. Assim, o algoritmo encontra soluções que estão, de certa forma, mais próximas dos pontos de referência escolhidos. Podemos observar como os pontos de referência podem influenciar a forma da fronteira de Pareto aproximada gerada na Fig. 16. Esta análise é

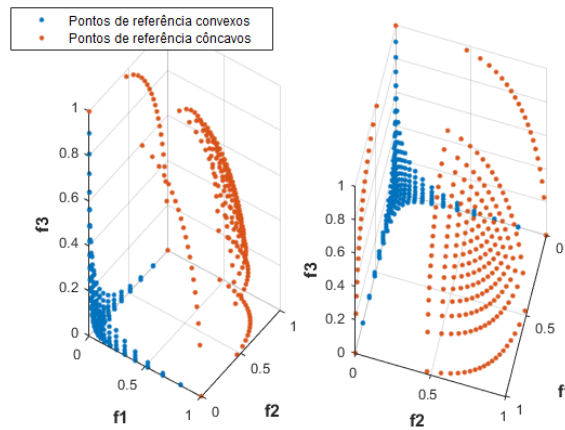


Figura 14 – Representação dos pontos de referência côncavo e convexos em um mesmo plano tridimensional.

realizada no problema de formato convexo, CDTLZ2.

A transformação Côncava apresentou maior número de pontos de referência no centro e gerou maior quantidade de soluções não dominadas na mesma região do PF (Fig. 14). Isso pode ser observado na Fig. 16(c). Com esse conjunto de pontos de referência, o NSGA-III realizou a busca em uma pequena região centralizada do FP. Portanto, para 3 funções objetivo, a Transformação Côncava, obtém não apenas soluções com alto valor de hiper-volume, mas centralizadas. No entanto, embora esta configuração de pontos tenha funcionado bem com WFG1 e WFG3 quando o número de objetivos aumentou, nos outros testes de benchmark teve um desempenho inferior aos outros algoritmos. A Fig. 22(c) sugeriu que este conjunto de pontos de referência não foi capaz de obter um conjunto diversificado de soluções não dominadas, uma vez que não conseguiu cobrir todas as soluções (LI; ZHEN; YAO, 2017).

Para a transformação Convexa, o conjunto de pontos de referência possui um maior número de pontos próximos à origem, como pode ser visto na Fig. 14. As soluções beneficiadas por este conjunto são aquelas mais próximas à origem. Portanto, este conjunto dá atenção especial às arestas do problema, embora ainda obtenha soluções não dominadas no meio da fronteira de Pareto, como mostra a Fig. 16(d). O algoritmo Convexo foi capaz de superar o NSGA-III em alguns problemas. Em problemas com três dimensões, NSGA-III (Fig. 16(j)) apresentou uma fronteira de Pareto bem dispersa sem qualquer agrupamento de soluções. Na Fig. 22(d) o algoritmo Convexo conseguiu uma melhor silhueta de soluções para objetivos próximos às bordas dos objetivos e no meio, quando comparado ao NSGA-III que concentrou suas soluções entre 0 e 0.4.

A transformação Senoidal gerou um conjunto de pontos de referência com algumas propriedades da transformação côncava e convexa. Portanto, ele pode obter soluções nas bordas como o conjunto convexo de pontos de referência e ainda ter uma quantidade certa de soluções próximas ao meio do PF como mostra a Fig. 16(e). Além disso, observe que foi possível obter

soluções no local exato dos morros, uma vez que o NSGA-III força que as soluções estejam associadas aos pontos de referências mais próximos dele, e usa isso como critério para que elas passem para a próxima geração. Embora quase não tivesse nenhuma diferença significativa do NSGA-III nos valores de hipervolume CDTLZ e DTLZ2, ele tinha uma fronteira de Pareto completamente diferente. Analisando a Fig. 22(e), é notável que o Algoritmo Senoidal tem suas soluções distribuídas mais uniformemente do que em comparação com NSGA-III.

4.2.4 Principais Conclusões

Esta capítulo apresenta o estudo da relação de influência entre a forma dos pontos de referência para o aprimoramento do desempenho de um algoritmo em MaOPs. São propostos três novos algoritmos que realizam a modificação do hiperplano utilizado pelo NSGA-III. Além disso, ele também usa um Algoritmo Guiado por Vetor de Referência, que de vez em quando transforma o Vetor de Referência de forma que cada Ponto de Referência em cada subespaço possa fazer uma contribuição igual para a população.

Experimentos mostraram que as transformações nos pontos de referência foram capazes de obter resultados que às vezes superam o NSGA-III. Isso confirma nossa hipótese, é possível melhorar o desempenho do NSGA-III usando pontos de referência em diferentes formas. As abordagens propostas também lidaram bem com a deterioração da pesquisa quando o número de objetivos aumentou e foram capazes de encontrar soluções bem convergentes e diversificadas.

Como um trabalho preliminar sobre a abordagem proposta, em (Oliveira; Britto, 2018) foi proposta uma adaptação do algoritmo NSGA-III usando quatro procedimentos para modificar a hipersuperfície e, portanto, os pontos de referência, que ocorrem logo após a etapa de normalização realizada por NSGA-III. Isso muda a forma como o NSGA-III associa e aplica a preservação de nicho. Três novas transformações, chamadas de convexa, côncava e senoidal e outra já explorada na literatura (a utilizada pelo RVEA), são comparadas com a versão original do NSGA-III. Os experimentos conduzidos em (Oliveira; Britto, 2018) mostraram que dependendo da transformação do ponto de referência, o desempenho do NSGA-III pode ser melhorado ou degenerado. Além disso, os autores concluíram que não é fácil avaliar a relação entre a forma da hiperplano obtida a partir da transformação dos pontos de referência e a Frente de Pareto. As análises realizadas em (Oliveira; Britto, 2018) também concluíram que a variante côncava NSGA-III pode encontrar um conjunto com boa convergência (devido ao valor de hipervolume competitivo) e diversificado de soluções para problemas envolvendo frentes de Pareto que têm uma densidade convexa, côncava, desarticulada e enviesada de pontos na frente e frentes com escalas diferentes, como em (HUBAND et al., 2006a). No entanto, apesar de que a modificação dos pontos de referência pode influenciar a busca, não há nenhum grande vencedor nesses algoritmos. Somente é possível observar que métodos diferentes obtêm bons resultados em diferentes cenários. Assim a ideia de continuação desse estudo consiste de criar um algoritmo capaz de alterar a hipersuperfície à medida que a busca evolui, criando desta maneira, um

algoritmo capaz de inferir qual a melhor hipersuperfície para cada tipo de problema.

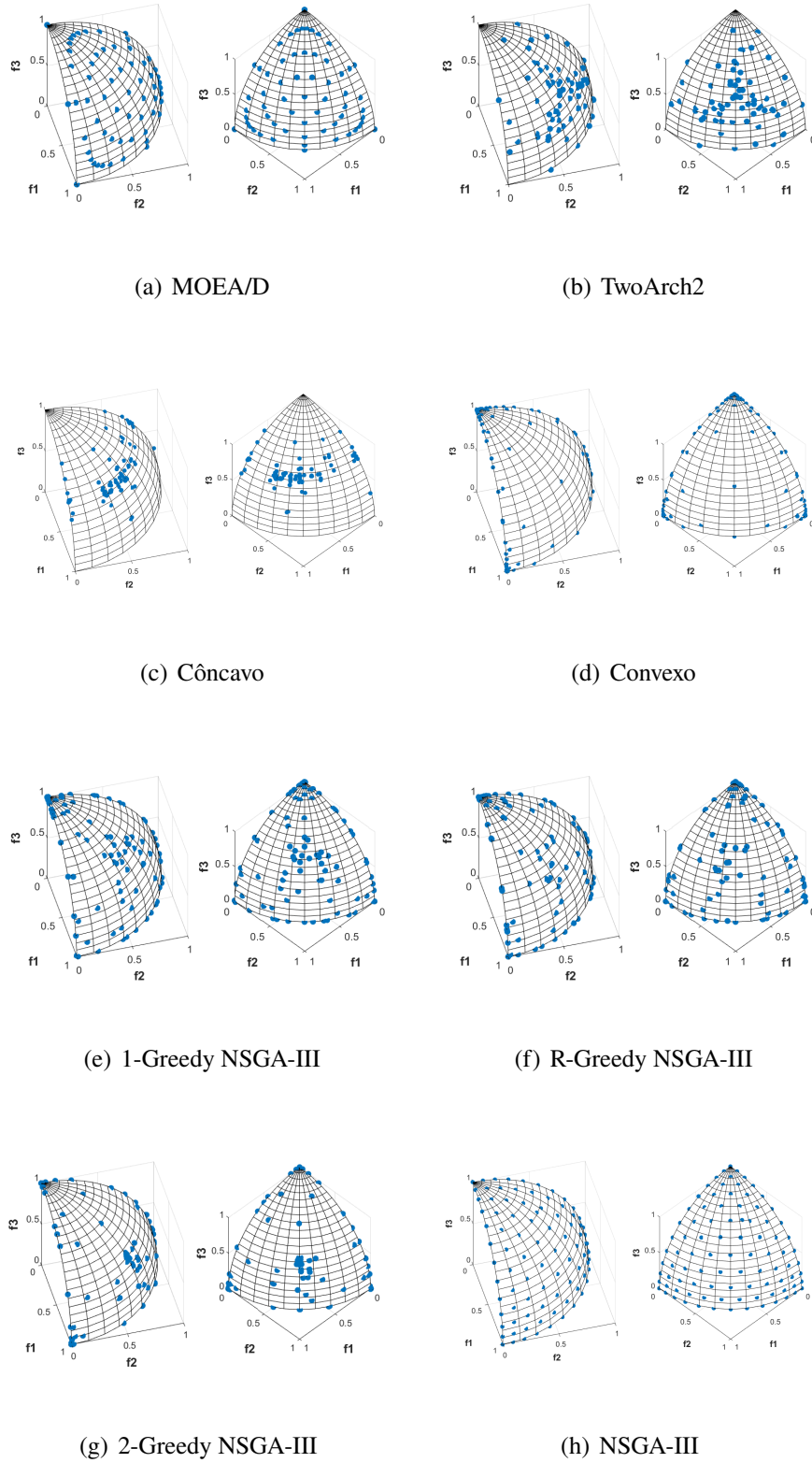


Figura 15 – As soluções não dominadas obtidas por cada algoritmo no DTLZ2 com 3 objetivos na melhor execução associada ao valor do hipervolume normalizado.

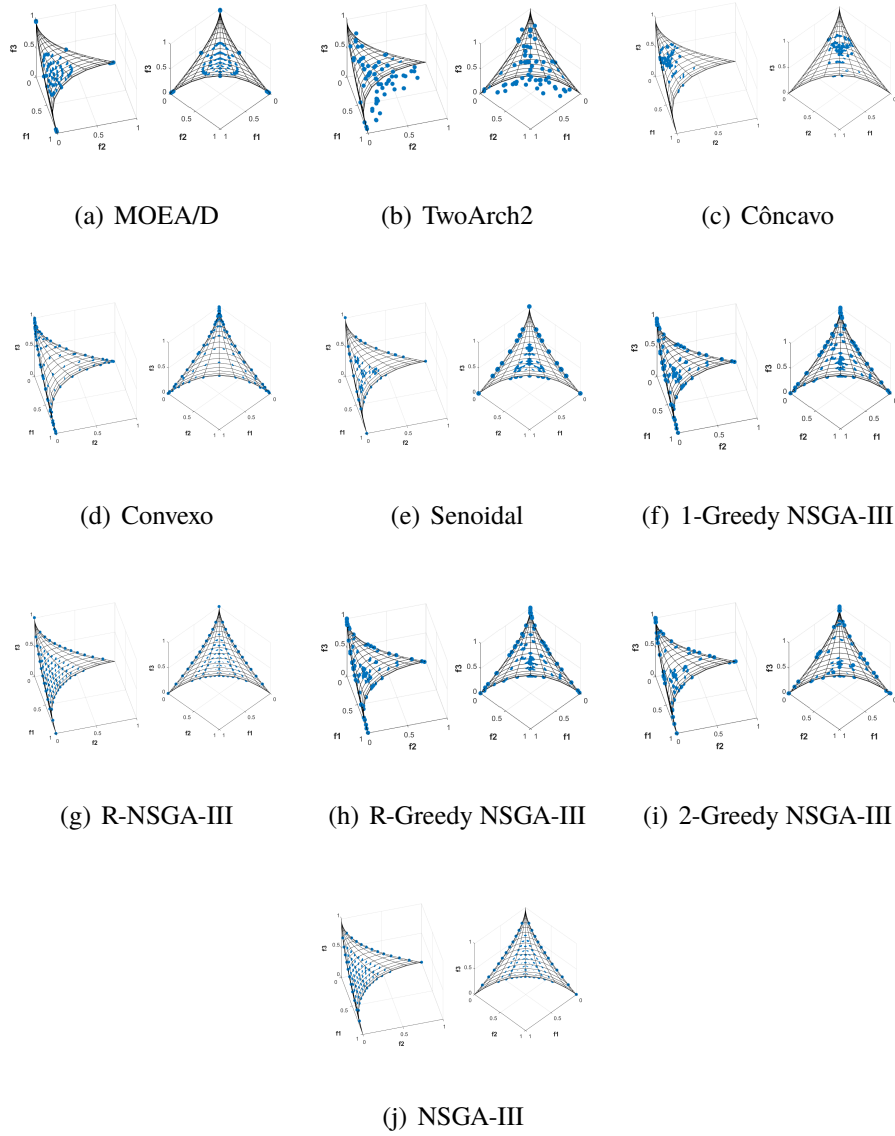


Figura 16 – As soluções não dominadas obtidas por cada algoritmo no CDTLZ2 com 3 objetivos na melhor execução associada ao valor do hipervolume normalizado.

5

Algoritmo K-Greedy: transformações conjugadas do hiperplano do NSGA-III

Como comentado no capítulo final anterior, apesar do autor original do NSGA-III ([DEB; JAIN, 2014](#)) saber que a distribuição dos pontos de referência pode afetar a distribuição das soluções, e consequentemente afetar o desempenho do algoritmo, nenhum estudo foi feito para avaliar a influência dos pontos de referência e a relação deles com a melhoria dos resultados de um MOEA.

Dessa forma, o K-Greedy tem o intuito de utilizar técnicas que explorem a relação dos pontos de referência e a melhoria na busca dos algoritmos de otimização. Para fazer isso, essa proposta explora uma abordagem gulosa no processo de alteração de pontos de referência, que consiste de realizar transformações matemáticas em alguns pontos com determinada frequência, avaliar a melhoria dessas transformações e mantê-las caso sejam benéficas.

Os algoritmos de otimização lidam com uma sequência de etapas, com cada etapa tendo um conjunto de escolhas/opções. Estratégias gulosas aparecem como alternativas interessantes para resolver esses problemas de tomada de decisão, pois escolhem a opção que parece ser a melhor no momento (escolha ideal local), esperando a sequência de soluções locais ideais ofereçam uma ótima solução global ([BRASSARD; BRATLEY, 1996](#)). Embora nem sempre seja possível encontrar uma solução ideal usando algoritmos gulosos, eles são eficientes em uma ampla variedade de problemas. Nesta seção, apresentamos a abordagem proposta, que abrange uma abordagem gulosa para decidir se aceita ou não as modificações executadas em parte do conjunto atual de pontos de referência. Em outras palavras, ele aceita as modificações apenas quando elas fornecem um incremento no hipervolume obtido usando os pontos de referência modificados. Dessa forma, o algoritmo proposto neste trabalho comporta-se como uma estratégia de escalada de montanhas, aceitando a orientação de novos pontos de referência somente quando eles movem a busca em direção às colinas mais altas da superfície do hipervolume.

De maneira resumida, o algoritmo proposto consiste em uma sequência de transformações matemáticas que são aplicadas aleatoriamente em um subconjunto do conjunto de pontos de

referência. A transformação fornece um novo conjunto de pontos de referência para a próxima geração apenas se o valor do hipervolume da solução aumentar, caso contrário, permanecerá o mesmo da geração anterior.

5.1 Algoritmo K-Greedy MaOEA

O K-Greedy é uma extensão do NSGA-III, e funciona como o NSGA-III visto no Fluxograma da Figura 1 em todas as etapas, com exceção da etapa de Associate, tendo em vista que os pontos de referência que compõem a hipersuperfície sofrem uma leve transformação antes de chegar nessa etapa, e dessa maneira acaba modificando a maneira com que os pontos de referência são associados e a etapa de *Niching*.

O Algoritmo 4 resume as principais etapas executadas pela abordagem denominada K-Greedy MaOEA, que se trata um algoritmo evolutivo de muitos objetivos baseado em indicadores gulosos e em dominância de Pareto, que usa uma sequência (com um máximo de K repetições) de transformações matemáticas específicas a serem aplicadas em um subconjunto de pontos de referência. Alternando a seleção entre diferentes transformações matemáticas e pontos de referência durante a pesquisa, o algoritmo parece ter uma melhor capacidade de ajustar o processo de otimização para lidar com diferentes fronteiras de Pareto que podem surgir, devidos aos diferentes problemas de Benchmark.

Algorithm 4 Pseudocódigo do algoritmo K-Greedy MaOEA usando uma sequência de até K transformações matemáticas por ponto de referência

INPUT: O conjunto de pontos de referência do NSGA-III $R_{NSGA_{III}} = \{\mathbf{r}_1, \mathbf{r}_1, \dots, \mathbf{r}_H\}$, o conjunto \mathfrak{M} de todas as transformações matemáticas, a frequência F_r , o número de pontos de transformações matemáticas simultâneas aplicadas K , o número de pontos de referência a serem alterados N_r , o novo conjunto de pontos de referência R^{new} , e o número total de iterações T .

```

1:  $t \leftarrow 0$ 
2:  $R^t = \{\mathbf{r}_1^t, \mathbf{r}_1^t, \dots, \mathbf{r}_H^t\} \leftarrow R_{NSGA_{III}} = \{\mathbf{r}_1, \mathbf{r}_1, \dots, \mathbf{r}_H\}$ 
3: for  $t \leftarrow 1$  to  $T$  do
4:   if  $(t \bmod F_r) = 0$  then
5:      $R^{new} \leftarrow R^t$ 
6:     for  $n \leftarrow 1$  to  $N_r$  do
7:       Escolha uma transformação matemática  $\mathcal{M}$  de  $\mathfrak{M}$ 
8:       Selecione um elemento  $\mathbf{r}_i^t$  do conjunto de pontos de referência atual  $R^t$ 
9:        $\mathbf{r}' \leftarrow \mathbf{r}_i^t$ 
10:      Aplique a transformação matemática  $\mathbf{r}' \leftarrow \mathcal{M}(\mathbf{r}')$  até  $K$  vezes
11:       $\mathbf{r}_i^{new} \leftarrow \mathbf{r}'$ 
12:    end for
13:    if  $HV(R^{new}) > HV(R^t)$  then
14:       $R^{t+1} \leftarrow R^{new}$ 
15:    else
16:       $R^{t+1} \leftarrow R^t$ 
17:    end if
18:  end if
19:   $t \leftarrow t + 1$ 
20:  Use o conjunto de pontos de referência  $R^t$  como usual no NSGA-III
21: end for

```

Considerando os parâmetros básicos para gerar o hiperplano discutidos na Seção 2, m como o número de objetivos, p como o número de divisões em cada dimensão do espaço de objetivo, e H , o número de pontos de referência, em função de m e p , o Algoritmo 4 começa inicializando o conjunto de pontos de referência atual $R^t = \{\mathbf{r}_1^t, \mathbf{r}_1^t, \dots, \mathbf{r}_H^t\}$ onde o elemento \mathbf{r}_i^t representa o i – simo ponto de referência no espaço de objetivos i.e. $\mathbf{r}_i^t = (r_1^t, \dots, r_m^t)$. Para isso, utilizam-se inicialmente os mesmos valores do conjunto adotado pelo NSGA-III. Então, o algoritmo entra no laço principal, num total de T repetições ou iterações. E a cada iteração t , se for o momento para atualizar os pontos de referência (um teste condicional é realizado na etapa 4 cujo resultado depende do parâmetro F_r), o Algoritmo 4 inicia o bloco de *transformações matemáticas* (do passo 6 ao passo 12). Caso contrário, o algoritmo incrementa o contador de iteração e segue as etapas tradicionais do NSGA-III usando o conjunto atual de pontos de referência.

No bloco de *transformações matemáticas*, o algoritmo seleciona um subconjunto de pontos de referência $R^t = \{\mathbf{r}_1^t, \mathbf{r}_1^t, \dots, \mathbf{r}_H^t\}$. Para cada ponto de referência \mathbf{r}' escolhido no passo anterior, o algoritmo seleciona uma transformação matemática \mathcal{M} (para os experimentos dessa dissertação, foram adotados somente côncavo e convexo, por se tratar de operações inversas) do conjunto de transformações matemáticas \mathfrak{M} e aplica no ponto de referência selecionado \mathbf{r}' . É

importante ressaltar que a seleção proposta nas etapas 7 e 8 do Algoritmo 4 pode considerar diferentes estratégias (da mais simples de todas, como uma seleção randômica, para as mais sofisticadas como aquelas utilizadas em hiperqueurísticas). Dependendo do valor do parâmetro K , a transformação matemática pode ser reaplicada (repetida) ou não. Exemplos de pontos de referências obtidos através da transformação matemática em todos os $N_r = 16$ pontos selecionados usando $K = 1$ (1-greedy) e $K = 2$ (2-greedy) podem ser vistos nas Figuras 17 e 18, respectivamente. Observe que na versão 2-greedy repete-se até duas vezes a transformação matemática em cada ponto selecionado, o que pode prover um conjunto de pontos de referência totalmente diferente, quando comparado a versão 1-greedy.

A Figura 19 mostra imagens de como o hiperplano evoluiu a medida que o número de gerações aumentou. O algoritmo sempre começa igualmente distribuído para todos os objetivos, mas é possível ver na Figura 19(a) que apenas com 50 gerações já foi possível encontrar um plano um pouco diferente, de 50 até 100 gerações o hiperplano se manteve igual, desta forma todas as alternativas foram descartadas, com 300 gerações o hiperplano já está sensivelmente modificado. Essas modificações continuam ocorrendo até a Figura 19(f), onde a maioria dos pontos já se encontram fora do plano e o algoritmo encontra seu critério de parada (o número de gerações).

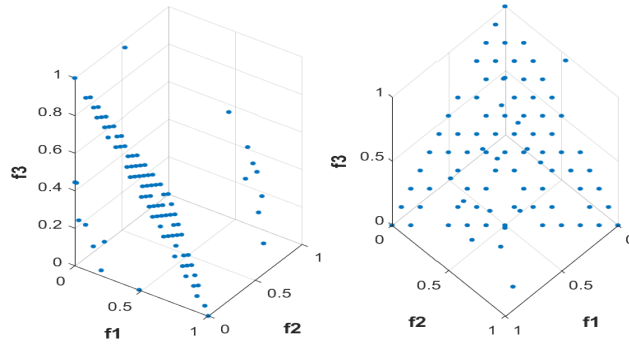


Figura 17 – 1-Greedy-MaOEA: conjunto R^{new} obtido após a aplicação de até uma única transformação matemática

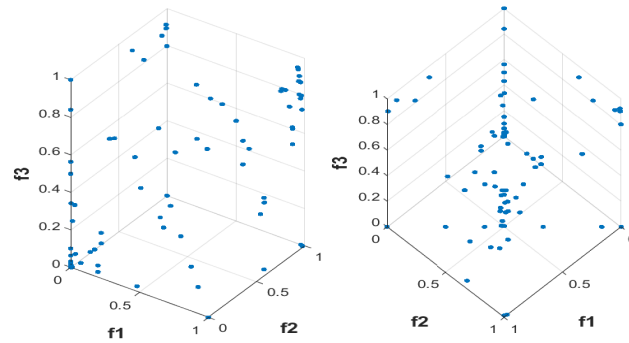


Figura 18 – 2-Greedy-MaOEA: conjunto R^{new} obtido após a aplicação de até duas transformações matemáticas

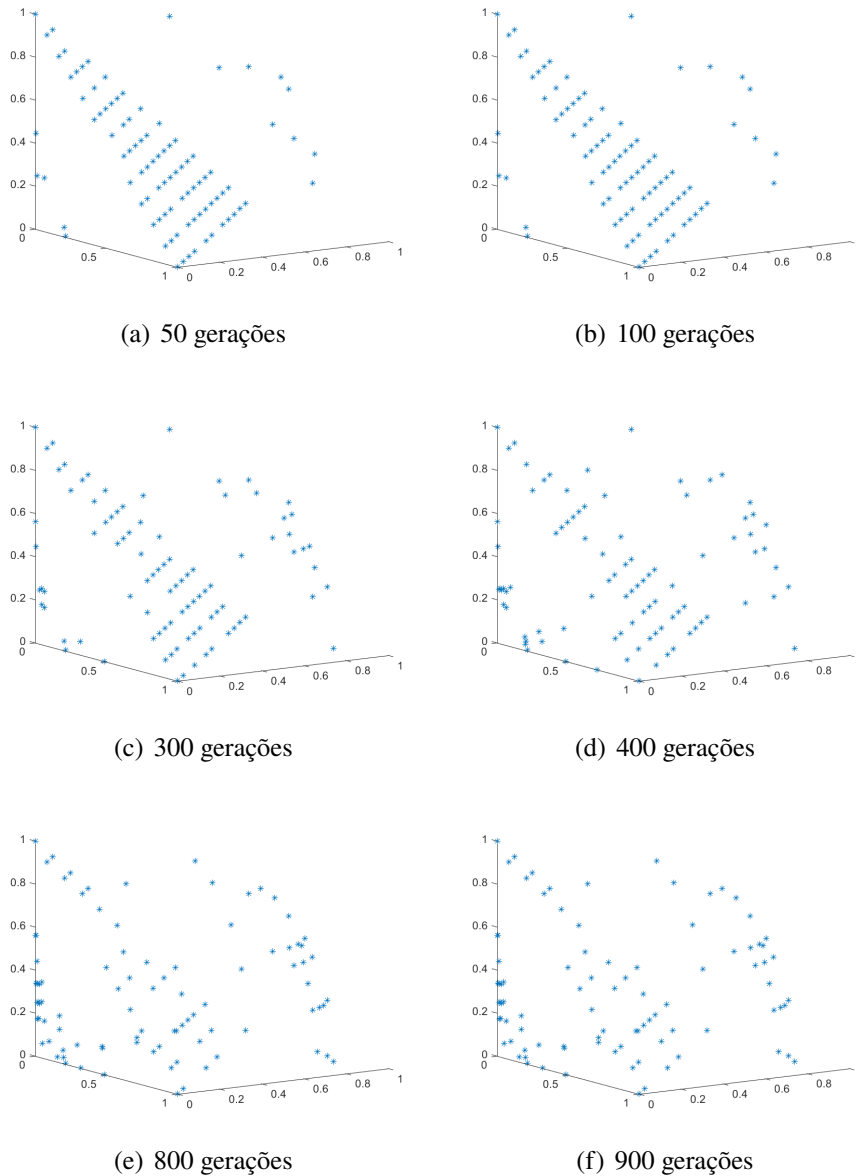


Figura 19 – Evolução da distribuição dos pontos para o Algoritmo *I-Greedy*

5.2 Metodologia

5.2.1 Problemas de Benchmark

Problemas de Benchmark são problemas parametrizáveis responsáveis por medir a qualidade de um algoritmo proposto. Esses problemas são escaláveis em complexidade e número de objetivos. Neles, é possível forçar certas características, como: *bias*, multi-modalidade, fronteiras desconexas ou conexas, geometrias convexas, concavas ou mistas, linearidade. Os problemas mais conhecidos da literatura são os da família DTLZ (DEB et al., 2002d) e WFG (HUBAND et al., 2006a) como visto na Figura 9.

Nos experimentos discutidos nessa seção, são usados 12 problemas diferentes sendo 7 da

família WFG (HUBAND et al., 2006a) e 5 da família DTLZ (DEB et al., 2002c). A família de problemas DTLZ pode ser escalada para qualquer número de objetivos (m) e variáveis de decisão (n). Além disso, é possível conhecer a Fronteira de Pareto para cada problema. A variável k representa a complexidade da busca, onde $k = n - m + 1$. Aqui, são usados os problemas DTLZ1 ao DTLZ4. O valor de k foi definido como $k = 10$, exceto para o DTLZ1, onde $k = 5$. Além disso foi utilizado o CDTLZ2.

Todos os problemas do DTLZ tratados neste documento são escaláveis com relação ao número de variáveis e funções de objetivo. O DTLZ1, tem uma estrutura linear e, à primeira vista, parece o hiperplano de referência definido pelo NSGA-III. Os problemas DTLZ2 a DTLZ4 têm uma forma côncava, a principal diferença é que no DTLZ2 e no DTLZ3 a densidade de soluções está uniformemente distribuída, por outro lado o DTLZ4 tem densidade de pontos polarizada (com as soluções favorecendo um lado da fronteira de Pareto). O DTLZ2 e o DTLZ4 são unimodais. Por outro lado, DTLZ1 e DTLZ3 são multimodais, o que é mais difícil de resolver, já que de acordo com Deb (DEB, 1999) ele pode ficar preso em ótimos locais. Em todos esses problemas DTLZ tratados neste trabalho, os objetivos podem ser classificados como sendo separáveis, isso significa que, se o algoritmo tentar otimizar um parâmetro de cada vez, ele identificará pelo menos um ponto ótimo global (HUBAND et al., 2006b). Olhando para essas características é possível observar que, o DTLZ4 é uma extensão do DTLZ2, e pode ser usado para estudar a capacidade de manter uma boa distribuição de soluções. O CDTLZ2 é uma versão convexa do problema de teste do DTLZ2, com as mesmas propriedades de ser separável e unimodal.

Os problemas da família WFG têm três parâmetros: um relacionado a posição, outro relacionado a distância e número de objetivos. Eles também são dimensionados para qualquer número de objetivos (m) e variáveis de decisão (D). Conforme sugerido em (CHENG et al., 2016), o número de variáveis de decisão pode ser definido como $D = c + l - 1$, onde l é a variável relacionada à distância. O valor de $l = 10$ foi usado em todos os problemas e a variável relacionada à posição $c = 4, 7$ e 9 foi usada para problemas com $m = 3, 8$ e 10 , respectivamente. Consideramos os problemas de instância WFG1 a WFG7 cuja escalabilidade, conhecimento do formato e localização do PF permanecem como no DTLZ. Cada algoritmo tem certas características. Como a PF é conhecida, é possível observar certos aspectos dos algoritmos e investigar seu desempenho enquanto o número de variáveis e dimensões aumenta.

Diferentemente dos problemas da família DTLZ, que possuem uma complexidade de pesquisa igual a $k = n - m + 1$, os problemas de teste do WFG podem ser manipulados por meio de um conjunto de transformações cujos recursos são apresentados no problema de teste. O WFG1 é projetado com uma distribuição regular e uma estrutura mista em sua PF, com uma estrutura também um pouco convexa. O WFG2 também possui uma forma convexa, mas nesse problema a sua fronteira está desconectada, o que significa que é esperado que haja grandes variações no valor do hipervolume. O WFG3 possui uma PF degenerada e linear e as variáveis de decisão

não são separáveis, isto significa que cada parâmetro não é independente do outro, portanto, não é possível otimizar considerando apenas um parâmetro por vez. WFG4 ao WFG7 tem a característica de ter uma PF côncava, eles também são projetados com diferentes dificuldades. O WFG4 possui multimodalidade, o WFG5 é enganoso, o WFG6 não é separável e o WFG7 é separável e unimodal.

5.2.2 Indicador de Qualidade e Visualização dos Resultados

Na literatura, diferentes indicadores de qualidade são usados para avaliar MOEAs. Embora não haja um acordo completo sobre quais métricas devem ser usadas, o hipervolume (HV) é o indicador de qualidade mais usado para avaliar MaOPs (Riquelme; Von Lücken; Baran, 2015). Em MOEAs baseados em indicadores, o indicador transforma o problema de otimização multiobjetivo em um único objetivo, onde o único critério é otimizar o próprio valor do indicador. Em vez de otimizar as funções objetivo diretamente, os algoritmos baseados em HV visam encontrar um conjunto de soluções que maximize o valor HV subjacente (BADER, 2010). O hipervolume não requer informação prévia sobre o verdadeiro ótimo de Pareto, é apenas necessário definir um ponto nadir, y^* . A Figura 20 ilustra quatro soluções não dominadas de uma fronteira de Pareto 2D (dois objetivos y_1 e y_2) em um problema de minimização e seu hipervolume associado (área cinza). Valores maiores de HV indicam melhores resultados em termos de convergência e diversidade no espaço objetivo.

Na literatura, diferentes indicadores de qualidade são usados para avaliar MOEAs. Para investigar o comportamento das abordagens propostas em termos de diversidade e convergência a PF, e o efeito da atual estratégia de pontos de referência utilizada é usado o HV. O HV é um dos mais usados indicadores de qualidade na Otimização com Muitos Objetivos, ele calcula o volume de todos os blocos retangulares a um dado ponto nadir. Para os pontos nadir que vão ser posteriormente propostos, o HV é um indicador de maximização, logo quanto maior o valor do HV um algoritmo tem, melhor a convergência e diversidade dessa resposta.

Para calcular o hipervolume, é necessário selecionar um ponto nadir como discutido anteriormente. Neste trabalho, $y^* = [1.5, 1.5, \dots, 1.5]$ é usado para DTLZ1; $y^* = [2, 2, \dots, 2]$ é usado para DTLZ2 e DTLZ4; $y^* = [11, 11, \dots, 11]$ é usado para DTLZ3 de acordo com (DEB et al., 2002c); e $y^* = [3, 5, \dots, 2m + 1]$ é usado para WFG1 a WFG7. Todos os valores de hipervolume apresentados neste trabalho são normalizados entre $[0, 1]$ dividindo POR $\prod_{i=1}^m y_i^*$, a mesma metodologia adotada por (CHENG et al., 2016). O algoritmo WFG Hypervolume (WHILE; BRADSTREET; BARONE, 2012) é usado para calcular o hipervolume, uma vez que o WFG é substancialmente mais rápido (especialmente quando $M \geq 5$) do que todos os algoritmos descritos anteriormente que calculam exatamente o hipervolume.

O experimento considera um total de execuções de 30 para cada algoritmo. Para realizar uma comparação justa, para cada problema, cada execução de algoritmo é limitada ao mesmo número de avaliações de funções objetivo. Como o objetivo deste estudo é avaliar a influência da

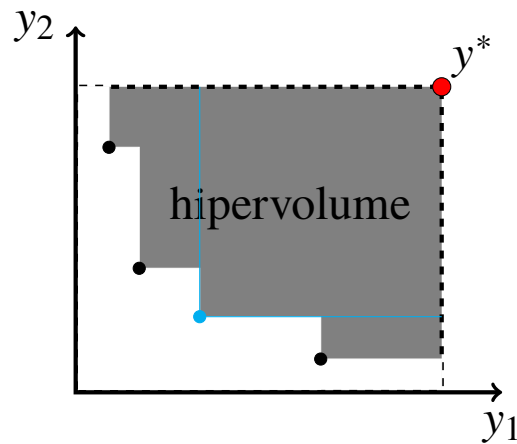


Figura 20 – Cálculo do hipervolume para uma Fronteira de Pareto de 2 objetivos: o ponto nadir (em vermelho) e quatro soluções não dominadas (com uma delas em azul com sua contribuição específica para o cálculo do hipervolume).

abordagem proposta, utilizando como base o NSGA-III, o teste de Wilcoxon é aplicado. O teste de Wilcoxon no nível de significância de 5% é adotado para comparar esses valores de hipervolume. O teste de Wilcoxon indica se um conjunto de dados fornecido tem alguma diferença significativa em relação ao conjunto de dados analisado. Em seguida, os valores médios são usados para decidir qual algoritmo tem o melhor valor. O teste de Wilcoxon está disponível no programa estatístico R ([R Core Team, 2020](#)) no pacote *Wilcox.test*. Nesta comparação, todas as variantes da abordagem proposta são comparadas com o algoritmo NSGA-III original (tradicional).

Em uma segunda etapa também é realizado o teste de Kruskal Wallis, com o intuito de comparar todos os algoritmos desenvolvidos nessa segunda etapa da dissertação entre si, de maneira paralela, tendo em vista que o teste de Kruskal Wallis se trata de um teste estatístico não paramétrico (que não necessita de uma distribuição modelo) e que deve ser feito para 3 ou mais grupos. O teste de Kruskal Wallis indica, com um certo nível de significância (5% no caso dos experimentos deste trabalho) se os valores do hipervolume possuem alguma diferença significativa. Nessa comparação foram utilizados os algoritmos da família K-GreedyMaOEA, TwoArch, NSGA-III e MOEA/DD. O teste de Kruskal Wallis também está disponível no programa estatístico R ([R Core Team, 2020](#)).

Além da análise de hipervolume, esta seção também discute a Fronteira de Pareto gerada. Na análise realizada, apenas CDTLZ2 e DTLZ2 são selecionados devido ao seu formato oposto. Gerou-se a visualização para objetivos de 3 e 10. Para a análise dos problemas com 3 objetivos a Fronteira de Pareto gerada foi ilustrada em um espaço dimensional de 3, para cada algoritmo. Para analisar os objetivos de 10, consideramos a estratégia de coordenadas paralelas. Para ilustrar o gráfico, selecionamos as soluções geradas pela execução com o melhor hipervolume.

5.2.3 Algoritmos e Parâmetros

Três novas variantes da abordagem proposta são avaliadas nesta seção: *1-Greedy*, *2-Greedy* e *R-Greedy*, em que "R" representa o procedimento de transformação do RVEA e "2" representa a possibilidade de aplicar duas vezes a transformação matemática realizada em cada ponto de referência. Apesar da possibilidade de usar estratégias mais sofisticadas, nos experimentos consideramos uma escolha aleatória e uniforme para selecionar os (N_r) pontos de referência e a transformação $\mathcal{M} \in \mathfrak{M}$.

As 5 variantes da abordagem proposta consideradas nos experimentos são:

- *Côncavo* (NSGA-III com a transformação Côncava): $|\mathfrak{M}| = 1$ e $\mathcal{M} \in \mathfrak{M}$ fixado como uma transformação côncava;
- *Convexa* (NSGA-III com a transformação Convexa): $|\mathfrak{M}| = 1$ e $\mathcal{M} \in \mathfrak{M}$ fixado como uma transformação convexa;
- *1-Greedy* (Estratégia Greedy e NSGA-III): $K = 1$ (a transformação matemática pode ser aplicada apenas 1 vez em cada ponto de referência), $|\mathfrak{M}| = 2$ e $\mathcal{M} \in \mathfrak{M}$ pode alternar entre transformações matemáticas côncavas e convexas;
- *2-Greedy* (Estratégia Greedy e NSGA-III): $K = 2$ (as transformações matemáticas podem ser aplicadas até duas vezes em cada ponto de referência), $|\mathfrak{M}| = 2$ e $\mathcal{M} \in \mathfrak{M}$ pode alternar entre transformações matemáticas côncavas e convexas;
- *R-Greedy* (NSGA-III com a estratégia 1-Greedy e RVEA): $K = 1$ (a transformação matemática pode ser aplicada apenas 1 vez em cada ponto de referência), $|\mathfrak{M}| = 3$ e $\mathcal{M} \in \mathfrak{M}$ pode trocar entre transformações matemáticas côncavas e convexas e também aplica a normalização do RVEA.

Além disso, como visto na Seção 2, um dos algoritmos mais utilizados na área é o MOEA/D, desta maneira decidiu-se também comparar os resultados da abordagem proposta com o MOEA/D. Para comparar a proposta com alguma estratégia baseada em arquivos, os algoritmos anteriores também são comparados com o TwoArch2.

Para conseguir uma comparação justa entre diferentes algoritmos, é necessário definir alguns parâmetros primeiro. Uma vez que este trabalho é baseado principalmente em (DEB; JAIN, 2014) e (CHENG et al., 2016), os mesmos conjuntos de parâmetros são adotados como nos artigos citados, para TwoArch2, são usados os mesmos parâmetros apenas quando é possível, todos os parâmetros relacionados ao TwoArch2 são os mesmos usados em (Wang; Jiao; Yao, 2015).

5.3 Resultados e Discussão

A Tabela 3 apresenta os resultados do hipervolume para todos os MaOPs analisados em 3 diferentes números de funções objetivo. Cada célula representa o valor médio do hipervolume para um número específico de funções objetivo. Como cada algoritmo é confrontado com o NSGA-III, os símbolos \approx , $+$ e $-$ indicam se a transformação é, respectivamente, significativamente igual, melhor ou pior que NSGA- III. Uma comparação é considerada igual se p - valor $> 0,05$, melhor se p - valor $\leq 0,05$ com um valor médio mais alto de hipervolume e pior se essa condição for falsa.

Tabela 3 – Resultados do Hipervolume. Comparação Principal Wilcoxon Rank Sum Test

Prob.	Obj.	MOEA/D	TwoArch2	RVEA	Côncavo	Convexo	1-Greedy	R-Greedy	2-Greedy	NSGA-III
CDTLZ2	3	0.989 (0.000)–	0.319 (0.044)–	0.993 (0.000)≈	0.993 (0.000)–	0.992 (0.000)–	0.992 (0.000)–	0.992 (0.000)–	0.990 (0.000)–	0.993 (0.000)
	8	1.000 (0.000)≈	0.639 (0.072)–	1.000 (0.000)≈	0.855 (0.069)–	0.999 (0.000)≈	0.999 (0.000)≈	1.000 (0.000)≈	1.000 (0.000)≈	1.000 (0.000)
	10	1.000 (0.000)≈	0.776 (0.051)–	1.000 (0.000)≈	0.877 (0.078)–	1.000 (0.000)≈	0.999 (0.000)≈	1.000 (0.000)≈	1.000 (0.000)≈	1.000 (0.000)
DTLZ1	3	0.989 (0.000)–	1.000 (0.000)+	0.992 (0.000)≈	0.802 (0.043)–	0.990 (0.000)≈	0.991 (0.000)–	0.989 (0.010)≈	0.990 (0.000)–	0.992 (0.000)
	8	0.998 (0.000)+	1.000 (0.000)+	0.982 (0.060)≈	0.821 (0.123)–	0.988 (0.044)≈	0.935 (0.097)≈	0.969 (0.045)≈	1.000 (0.000)+	0.972 (0.025)
	10	0.997 (0.001)–	1.000 (0.000)+	0.995 (0.020)≈	0.922 (0.042)–	0.998 (0.005)≈	0.984 (0.043)–	0.989 (0.024)–	0.999 (0.000)≈	0.999 (0.001)
DTLZ2	3	0.921 (0.000)–	0.342 (0.043)–	0.927 (0.000)≈	0.701 (0.031)–	0.919 (0.000)–	0.925 (0.000)–	0.923 (0.001)–	0.921 (0.001)–	0.927 (0.000)
	8	0.951 (0.009)–	0.589 (0.068)–	0.982 (0.024)–	0.678 (0.047)–	0.988 (0.015)≈	0.987 (0.014)≈	0.991 (0.017)+	0.983 (0.022)≈	0.986 (0.014)
	10	0.957 (0.008)–	0.777 (0.037)–	0.989 (0.010)≈	0.700 (0.050)–	0.989 (0.011)≈	0.989 (0.009)≈	0.987 (0.009)≈	0.987 (0.012)≈	0.986 (0.011)
DTLZ3	3	0.920 (0.000)–	1.000 (0.000)≈	1.000 (0.000)≈	0.951 (0.014)–	0.999 (0.001)≈	0.999 (0.000)–	0.999 (0.000)–	0.999 (0.000)–	1.000 (0.000)
	8	0.954 (0.011)–	1.000 (0.000)+	0.998 (0.005)≈	0.876 (0.122)–	1.000 (0.001)≈	0.996 (0.001)≈	0.999 (0.000)+	0.999 (0.002)+	0.991 (0.032)
	10	0.946 (0.013)–	1.000 (0.000)≈	1.000 (0.000)≈	0.943 (0.073)–	1.000 (0.000)≈	1.000 (0.000)≈	0.999 (0.000)≈	1.000 (0.000)+	0.999 (0.001)
DTLZ4	3	0.871 (0.051)+	0.331 (0.037)–	0.851 (0.064)–	0.834 (0.037)–	0.877 (0.059)+	0.840 (0.129)–	0.898 (0.051)+	0.863 (0.102)+	0.859 (0.133)
	8	0.939 (0.009)–	0.658 (0.063)–	0.997 (0.003)≈	0.857 (0.071)–	0.998 (0.002)≈	0.998 (0.001)+	0.998 (0.001)+	0.998 (0.001)+	0.997 (0.002)
	10	0.925 (0.012)–	1.000 (0.000)≈	1.000 (0.000)≈	0.879 (0.043)–	1.000 (0.000)≈	1.000 (0.000)≈	1.000 (0.000)≈	1.000 (0.000)≈	1.000 (0.000)
WFG1	3	0.876 (0.016)+	0.128 (0.103)–	0.811 (0.023)–	0.716 (0.044)–	0.855 (0.032)+	0.831 (0.015)≈	0.837 (0.030)+	0.819 (0.018)≈	0.816 (0.027)
	8	0.871 (0.073)+	0.016 (0.033)–	0.699 (0.030)≈	0.730 (0.027)+	0.715 (0.030)≈	0.716 (0.032)≈	0.731 (0.022)+	0.712 (0.028)≈	0.703 (0.020)
	10	0.912 (0.028)+	0.000 (0.000)–	0.713 (0.024)–	0.775 (0.025)+	0.713 (0.027)≈	0.713 (0.019)≈	0.724 (0.014)+	0.727 (0.027)+	0.709 (0.026)
WFG2	3	0.829 (0.067)–	0.662 (0.047)–	0.865 (0.078)–	0.583 (0.149)–	0.868 (0.073)–	0.895 (0.074)–	0.902 (0.070)–	0.873 (0.072)–	0.906 (0.073)
	8	0.973 (0.024)–	0.978 (0.009)–	0.996 (0.004)≈	0.933 (0.077)–	0.999 (0.001)+	0.997 (0.004)+	0.998 (0.031)+	0.998 (0.001)+	0.995 (0.004)
	10	0.984 (0.020)–	0.986 (0.007)–	0.999 (0.001)≈	0.981 (0.016)–	0.999 (0.000)≈	0.998 (0.001)≈	0.998 (0.001)≈	0.999 (0.000)≈	0.999 (0.002)
WFG3	3	0.712 (0.000)≈	0.434 (0.029)–	0.708 (0.003)–	0.475 (0.175)–	0.709 (0.004)–	0.707 (0.007)–	0.707 (0.013)–	0.705 (0.005)–	0.711 (0.003)
	8	0.466 (0.017)–	0.517 (0.064)–	0.620 (0.018)≈	0.624 (0.024)≈	0.645 (0.015)+	0.630 (0.021)≈	0.646 (0.014)+	0.614 (0.032)≈	0.619 (0.016)
	10	0.522 (0.027)–	0.503 (0.053)–	0.600 (0.025)≈	0.612 (0.059)+	0.651 (0.014)+	0.627 (0.020)+	0.644 (0.019)+	0.635 (0.014)+	0.602 (0.030)
WFG4	3	0.699 (0.000)–	0.326 (0.042)–	0.730 (0.000)≈	0.397 (0.077)–	0.707 (0.002)–	0.721 (0.003)–	0.715 (0.008)–	0.710 (0.003)–	0.730 (0.001)
	8	0.665 (0.005)–	0.580 (0.048)–	0.906 (0.067)≈	0.546 (0.103)–	0.905 (0.030)–	0.941 (0.002)+	0.938 (0.004)+	0.938 (0.001)+	0.929 (0.041)
	10	0.594 (0.085)–	0.785 (0.032)–	0.926 (0.047)≈	0.642 (0.095)–	0.938 (0.036)≈	0.954 (0.049)+	0.964 (0.024)+	0.906 (0.087)≈	0.897 (0.087)
WFG5	3	0.662 (0.005)–	0.294 (0.086)–	0.694 (0.003)≈	0.399 (0.073)–	0.672 (0.002)–	0.687 (0.004)–	0.680 (0.009)–	0.677 (0.005)–	0.694 (0.003)
	8	0.634 (0.000)–	0.578 (0.055)–	0.895 (0.000)≈	0.738 (0.038)–	0.861 (0.000)–	0.893 (0.001)–	0.888 (0.002)–	0.889 (0.002)–	0.895 (0.000)
	10	0.625 (0.043)–	0.785 (0.032)–	0.919 (0.000)≈	0.779 (0.048)–	0.902 (0.000)–	0.919 (0.000)≈	0.918 (0.000)–	0.916 (0.000)–	0.919 (0.000)
WFG6	3	0.650 (0.020)–	0.315 (0.096)–	0.683 (0.015)≈	0.344 (0.060)–	0.660 (0.015)–	0.676 (0.013)–	0.660 (0.012)–	0.662 (0.011)–	0.685 (0.012)
	8	0.312 (0.140)–	0.553 (0.056)–	0.871 (0.016)≈	0.453 (0.100)–	0.833 (0.016)–	0.870 (0.020)+	0.881 (0.019)+	0.866 (0.021)≈	0.869 (0.016)
	10	0.318 (0.154)–	0.750 (0.054)–	0.890 (0.030)≈	0.532 (0.099)–	0.891 (0.017)≈	0.888 (0.027)≈	0.894 (0.018)≈	0.889 (0.021)≈	0.886 (0.035)
WFG7	3	0.699 (0.000)–	0.202 (0.077)–	0.731 (0.000)≈	0.292 (0.050)–	0.705 (0.001)–	0.720 (0.003)–	0.713 (0.006)–	0.710 (0.004)–	0.731 (0.000)
	8	0.628 (0.052)–	0.560 (0.100)–	0.944 (0.001)≈	0.474 (0.122)–	0.905 (0.027)–	0.943 (0.001)+	0.941 (0.003)+	0.938 (0.002)+	0.936 (0.037)
	10	0.561 (0.072)–	0.713 (0.044)–	0.934 (0.033)≈	0.588 (0.086)–	0.936 (0.031)–	0.956 (0.026)≈	0.959 (0.028)≈	0.960 (0.024)+	0.953 (0.031)

Tabela 4 – Investigação da influência dos Parâmetros no Algoritmo R-Greedy . Nome do problema: WFG1

$F_r = 5$	N_r	Objetivos		
		3	8	10
	1	0.830 (0.022)≈	0.708 (0.030)≈	0.722 (0.025)+
	5	0.833 (0.028)≈	0.699 (0.034)≈	0.717 (0.030)≈
	10	0.842 (0.025)+	0.699 (0.025)≈	0.716 (0.030)≈
$F_r = 10$	N_r	Objetivos		
		3	8	10
	1	0.831 (0.032)≈	0.698 (0.039)≈	0.718 (0.020)+
	5	0.834 (0.032)≈	0.712 (0.030)≈	0.720 (0.025)+
	10	0.837 (0.026)+	0.697 (0.086)≈	0.711 (0.026)≈
$F_r = 25$	N_r	Objetivos		
		3	8	10
	1	0.823 (0.021)≈	0.721 (0.035)+	0.716 (0.026)+
	5	0.834 (0.025)+	0.7035 (0.027)≈	0.708 (0.027)≈
	10	0.828 (0.021)≈	0.707 (0.024)≈	0.710 (0.028)≈

Tabela 5 – Investigação da influência dos Parâmetros no Algoritmo R-Greedy . Nome do problema: WFG3

$F_r = 5$	N_r	Objetivos		
		3	8	10
	1	0.709 (0.004)≈	0.646 (0.017)+	0.621 (0.030)+
	5	0.707 (0.004)–	0.641 (0.025)+	0.651 (0.025)+
	10	0.696 (0.014)–	0.642 (0.025)+	0.658 (0.014)+
$F_r = 10$	N_r	Objetivos		
		3	8	10
	1	0.708 (0.005)≈	0.632 (0.002)+	0.613 (0.036)≈
	5	0.705 (0.005)–	0.632 (0.026)+	0.645 (0.024)+
	10	0.697 (0.010)–	0.642 (0.019)+	0.652 (0.019)+
$F_r = 25$	N_r	Objetivos		
		3	8	10
	1	0.711 (0.003)≈	0.622 (0.028)≈	0.605 (0.020)+
	5	0.709 (0.005)≈	0.644 (0.013)+	0.626 (0.028)+
	10	0.705 (0.007)–	0.640 (0.619)+	0.637 (0.024)+

5.3.1 Efeito do K-Greedy sobre o NSGA-III

Nos MaOPs geralmente os algoritmos não conseguem encontrar toda a fronteira de Pareto, porque o número de pontos de referência necessários para cobri-la é muito alto. Assim, um algoritmo que pode se concentrar em determinadas regiões pode ser interessante, pois há uma garantia de que pelo menos uma parte de interesse foi coberta pelo algoritmo.

A transformação côncava tem um número maior de pontos de referência no centro e gerou uma quantidade maior de soluções não dominadas na mesma região do PF (Fig. 12). Isso pode ser observado na Fig. 16(c). Com esse conjunto de pontos de referência, o NSGA-III realizou a busca em uma pequena região centralizada do PF. Portanto, para três funções objetivo, a Transformação Côncava obtém não apenas soluções com alto valor de hipervolume, mas centralizada. No entanto, embora essa configuração de pontos de referência tenha lidado bem com o WFG1 e o WFG3 a medida que o número de objetivos aumenta, para os outros testes de benchmark, ele apresenta um comportamento ruim. Fig. 22(c) e Fig. 21(c) sugeriram que esse conjunto de pontos de referência não é capaz de obter um conjunto diversificado de soluções não dominadas para esta instância como pode ser visto em Li, Zhen e Yao (2017) que trata sobre visualização em coordenadas paralelas.

Observando os resultados de cada transformação individualmente: a Transformação Côncava obteve os melhores resultados em termos de hipervolume contra o NSGA-III em 3 de 33 MaOPs. Pode-se notar que essas transformações têm melhores valores de hipervolumes do que os outros algoritmos no WFG1 e WFG3, especialmente quando se trata de um alto número de objetivos, mesmo apesar de o WFG1 ser convexo. Isso implica que o melhor conjunto de pontos de referência não tem necessariamente a mesma forma da fronteira de Pareto.

Para a transformação *Convexa*, o conjunto de pontos de referência possui um número maior de pontos perto da origem, como visto na Fig. 11. As soluções beneficiadas por este conjunto são aquelas que são mais próximas à origem. Portanto, este conjunto dá atenção especial às margens do problema, embora ainda tenha soluções não dominadas no meio da fronteira de Pareto, como mostra a Fig. 16(d). O algoritmo *Convex* pode superar o NSGA-III em alguns problemas. Em problemas tridimensionais, o NSGA-III (Fig. 16(j)) apresentou uma fronteira de Pareto bem dispersa, sem agrupamento de soluções. Na Fig. 22(d) e Fig. 21(d) o algoritmo *Convex* obteve soluções no meio da fronteira, e ainda assim obteve uma boa convergência, tendo em vista que conseguiu obter 1 em todas as soluções, por outro lado a maioria dos outros algoritmos ou tiveram silhuetas irregulares, que significa que as soluções não estão bem distribuídas, ou concentraram as soluções em alguma região específica, como o NSGA-III com a maioria das suas soluções entre 0 e 0.2.

Em geral, é possível influenciar a busca do algoritmo escolhendo os pontos de referência e melhorar os resultados do NSGA-III. No geral, o NSGA-III original sustentou os bons resultados em quase todas as comparações. No entanto, toda transformação proposta melhorou os resultados do NSGA-III em pelo menos um problema. Portanto, os pontos de referência são uma maneira promissora de melhorar os resultados do NSGA-III. Em especial destacam-se as variantes *1-Greedy*, *R-Greedy* e *2-Greedy*, que podem superar o NSGA-III em vários dos cenários ou obter resultados semelhantes quando o número de objetivos aumenta.

Observando os problemas da DTLZ, todos os algoritmos propostos, exceto os de formato côncavo, podem obter resultados tão bons quanto o NSGA-III. Embora DTLZ2, DTLZ3 e DTLZ4 sejam MaOPs de formato côncavo, a Transformação *Convexa* funcionou bem. O principal motivo é que um conjunto de pontos de referência em forma convexa tem uma alta densidade de pontos de referência perto da solução ideal 0. Por isso, atrai respostas mais próximas de uma região com maior retorno e para esse problemas se mostrou o melhor formato.

Nos problemas de DTLZ1 a DTLZ4, o melhor algoritmo não sofre muitas variações, tendo em vista que todos os algoritmos conseguem obter bons resultados, o que é muito interessante, pois os algoritmos têm fronteira de Pareto completamente diferentes, com o DTLZ1 sendo linear e o DLTZ2, DTLZ3 e DTLZ4 côncavos, com alguns deles tendo uma distribuição não homogênea de soluções. Por outro lado, o WFG1 a WFG7 mostra que o melhor algoritmo pode mudar, indicando uma alta influência dos pontos de referência. Esse resultado é mais evidente no WFG, onde alguns problemas se comportam bem quando há pontos de referência mais côncavos (como

WFG1, WFG3 e WFG5) e, em outros problemas, o melhor algoritmo é convexo ou alguma mistura dos dois.

Em geral, é possível observar R-Greedy tem um desempenho competitivo, enfatizando que, na maioria dos casos, a influência da modificação dos pontos é positiva ou simplesmente não há diferença estatística. Em alguns casos, R-Greedy tem resultados piores. Isso ocorre porque, em alguns problemas, o conjunto original de pontos pode ser melhor ou a amostragem R-Greedy não é grande o suficiente para gerar um bom conjunto de pontos de referência para o problema. Um exemplo é o DTLZ1, cujo a fronteira de Pareto tem o mesmo formato dos pontos originais. Assim, ao executar o R-Greedy, há desperdício de recursos tentando melhorar pontos que já são adequados para aproximar o PF. É importante notar que este comportamento é mais comum nas versões anteriores da abordagem proposta (ou seja, aquelas que usam apenas transformações Côncavas ou Convexas), pois não são capazes de se adaptar nestes casos, resultando em formas rígidas que acabam proporcionando resultados piores. Apesar de não superar o NSGA-III nesses casos específicos, o R-Greedy tem um desempenho melhor do que os algoritmos que consideram transformação matemática de forma isolada, pois suaviza a modificação dos pontos de referência, especialmente para valores de K mais altos.

Depois que um conjunto de soluções não dominadas é encontrado, a próxima pergunta pode ser como escolher a mais adequada para os tomadores de decisão? A tomada de decisão multicritério (MCDM) é um problema complexo (ODU, 2019; VELASQUEZ; HESTER, 2013) e frequentemente, depende das prioridades dos objetivos ou preferências definidas pelo tomador de decisão. O MCDM está fora do escopo desse trabalho, mas o tema pode ser mais explorado por estar vinculado ao conceito central da otimização, que pressupõe a possibilidade de distribuição não uniforme para o conjunto de pontos de referência e reforça a busca nas proximidades deles. Ao final do processo evolutivo, o tomador de decisão poderia, auxiliado por um processo de agrupamento, escolher um ou mais grupos de soluções não dominadas a serem examinados, reduzindo o desafiador procedimento de olhar para toda a fronteira. Assim, além das técnicas de decisão multicritério tradicionais (VELASQUEZ; HESTER, 2013), a distribuição não uniforme poderia ajudar na tomada de decisão final. Além disso, embora as preferências de objetivo não tenham sido exploradas neste trabalho, elas poderiam ser utilizadas, além das informações de melhoria do hipervolume, para orientar a busca pela transformação matemática aplicada aos pontos de referência selecionados.

5.3.2 Análise dos parâmetros F_r e N_r

Esta subseção discute se os parâmetros F_r e N_r podem ser problemáticos para ajustar, causando, portanto, um problema para encontrar os melhores parâmetros para cada problema.

1-Greedy e especialmente R-Greedy mostram alguns resultados interessantes ao lidar com problemas do WFG. É possível ver que, quando o número de funções objetivo aumenta na Tabela 3, ele começa a obter o melhor resultado em comparação com todos os algoritmos

anteriores e até a superar quando é possível, conforme pretendido.

Como com o R-Greedy é possível gerar todos as transformações matemáticas anteriores, tendo em vista que ele pode aplicar transformações concavas e convexas bem como aplicar ou não o RVEA. Foi feito um estudo para ver como os parâmetros F_r e N_r podem se relacionar com seus resultados. As Tabelas 4 e 5 mostram o comportamento do R-Greedy para 2 problemas de teste para os quais o NSGA-III original teve um desempenho ruim. As tabelas 4 e 5 não mostram um único caso uma derrota do algoritmo R-Greedy quando o número de objetivos aumenta. Como o número de pontos de referência cresce com o número de funções objetivo, espera-se que para $M = 10$ os melhores resultados sejam mais dependentes dos parâmetros, considerando que a proporção de pontos de referência modificados pelo número total de pontos de referência está deteriorada à medida que o número de funções objetivo aumenta. A melhor proporção da média sobre o desvio padrão está no canto inferior direito da matriz, indicando alguns parâmetros preferenciais. É justo supor que para $M = 10$ e $M = 8$ o melhor resultado não depende muito do parâmetro F_r , mas é possível ver uma tendência de melhoria aumentando o valor de N_r . Por outro lado, a diferença entre $N_r = 1$ a $N_r = 5$ é melhor que $N_r = 5$ a $N_r = 10$, portanto, o melhor valor de N_r é de cerca de 5% do número de pontos de referência. Também é possível ver uma tendência de melhoria dos resultados quando F_r diminui e N_r aumenta.

Os resultados 2-Greedy são particularmente interessantes, pois melhoram bastante quando o número de funções objetivo aumenta. A única desvantagem em 8 ou 10 objetivos aconteceu no problema WFG5, mas, observando a diferença geral, o valor é insignificante. Outro recurso bastante interessante pode ser observado nas Fig. 22(i) e 21(g), para os problemas DTLZ2 e CDLTZ2, o algoritmo 2-Greedy apresentou um conjunto diversificado de soluções não dominadas para esse exemplo, resolvendo algumas desvantagens do algoritmo anterior (R-Greedy e 1-Greedy) e mostrando uma maneira promissora de melhorar o comportamento do NSGA-III em problemas com muitos objetivos. Na Fig. 16(i) e Fig. 15(g) o algoritmo 2-Greedy concedeu uma boa silhueta de soluções cujos objetivos estavam nos extremos da fronteira, por outro lado, o NSGA-III é melhor em obter soluções uniformemente dispersas na PF devido a seus pontos de referência bem distribuídos.

5.3.3 Comparação contra algoritmos do estado da arte

Nesta subseção é apresentada uma comparação considerando os algoritmos propostos, NSGA-III e dois algoritmos do estado da arte. Os valores de hipervolume de MOEA/D e TwoArch2 também são apresentados na Tabela 3. O teste estatístico utilizado para comparação é o de Kruskal Wallis.

Quando o desempenho dos algoritmos da família K-Greedy são comparados com MOEA/D e TwoArch é possível observar que ele supera os dois algoritmos na maioria dos cenários, considerando a média HV e o desvio padrão. Para problemas da família DTLZ, MOEA/D tem desempenho sempre pior, exceto no caso de DTLZ1 para uma pequena diferença. TwoArch

apresenta o mesmo padrão, embora a diferença seja ainda menor. Observando os problemas da família WFG, nota-se que, para todos os problemas considerados, o TwoArch não supera os algoritmos propostos para nenhum objetivo. MOEA/D teve um desempenho ligeiramente melhor vencendo em 4 problemas diferentes (apenas um com 10 objetivos), particularmente WFG1, onde foi capaz de superar as abordagens propostas em todos os cenários.

Tabela 6 – Resultados do Hipervolume. Comparação Principal Kruskal Wallis

Problema	Objetivos	MOEAD	TwoArch2	1-Greedy	R-Greedy	2-Greedy	NSGA-III
CDTLZ2	3	0.989 (0.000)	0.319 (0.044)	0.992 (0.000)	0.992 (0.000)	0.990 (0.000)	0.993 (0.000)
	8	1.000 (0.000)	0.639 (0.072)	0.999 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
	10	1.000 (0.000)	0.776 (0.051)	0.999 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
DTLZ1	3	0.989 (0.000)	1.000 (0.000)	0.991 (0.000)	0.989 (0.010)	0.990 (0.000)	0.992 (0.000)
	8	0.998 (0.000)	1.000 (0.000)	0.935 (0.097)	0.969 (0.045)	1.000 (0.000)	0.972 (0.025)
	10	0.997 (0.001)	1.000 (0.000)	0.984 (0.043)	0.989 (0.024)	0.999 (0.000)	0.999 (0.001)
DTLZ2	3	0.921 (0.000)	0.342 (0.043)	0.925 (0.000)	0.923 (0.001)	0.921 (0.001)	0.927 (0.000)
	8	0.951 (0.009)	0.589 (0.068)	0.987 (0.014)	0.991 (0.017)	0.983 (0.022)	0.986 (0.014)
	10	0.957 (0.008)	0.777 (0.037)	0.989 (0.009)	0.987 (0.009)	0.987 (0.012)	0.986 (0.011)
DTLZ3	3	0.920 (0.000)	1.000 (0.000)	0.999 (0.000)	0.999 (0.000)	0.999 (0.000)	1.000 (0.000)
	8	0.954 (0.011)	1.000 (0.000)	0.996 (0.001)	0.999 (0.000)	0.999 (0.002)	0.991 (0.032)
	10	0.946 (0.013)	1.000 (0.000)	1.000 (0.000)	0.999 (0.000)	1.000 (0.000)	0.999 (0.001)
DTLZ4	3	0.871 (0.051)	0.331 (0.037)	0.840 (0.129)	0.898 (0.051)	0.863 (0.102)	0.859 (0.133)
	8	0.939 (0.009)	0.658 (0.063)	0.998 (0.001)	0.998 (0.001)	0.998 (0.001)	0.997 (0.002)
	10	0.925 (0.012)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
WFG1	3	0.876 (0.016)	0.128 (0.103)	0.831 (0.015)	0.837 (0.030)	0.819 (0.018)	0.816 (0.027)
	8	0.871 (0.073)	0.016 (0.033)	0.716 (0.032)	0.731 (0.022)	0.712 (0.028)	0.703 (0.020)
	10	0.912 (0.028)	0.000 (0.000)	0.713 (0.019)	0.724 (0.014)	0.727 (0.027)	0.709 (0.026)
WFG2	3	0.829 (0.067)	0.662 (0.047)	0.895 (0.074)	0.902 (0.070)	0.873 (0.072)	0.906 (0.073)
	8	0.973 (0.024)	0.978 (0.009)	0.997 (0.004)	0.998 (0.031)	0.998 (0.001)	0.995 (0.004)
	10	0.984 (0.020)	0.986 (0.007)	0.998 (0.001)	0.998 (0.001)	0.999 (0.000)	0.999 (0.002)
WFG3	3	0.712 (0.000)	0.434 (0.029)	0.707 (0.007)	0.707 (0.013)	0.705 (0.005)	0.711 (0.003)
	8	0.466 (0.017)	0.517 (0.064)	0.630 (0.021)	0.646 (0.014)	0.614 (0.032)	0.619 (0.016)
	10	0.522 (0.027)	0.503 (0.053)	0.627 (0.020)	0.644 (0.019)	0.635 (0.014)	0.602 (0.030)
WFG4	3	0.699 (0.000)	0.326 (0.042)	0.721 (0.003)	0.715 (0.008)	0.710 (0.003)	0.730 (0.001)
	8	0.665 (0.005)	0.580 (0.048)	0.941 (0.002)	0.938 (0.004)	0.938 (0.001)	0.929 (0.041)
	10	0.594 (0.085)	0.785 (0.032)	0.954 (0.049)	0.964 (0.024)	0.906 (0.087)	0.897 (0.087)
WFG5	3	0.662 (0.005)	0.294 (0.086)	0.687 (0.004)	0.680 (0.009)	0.677 (0.005)	0.694 (0.003)
	8	0.634 (0.000)	0.578 (0.055)	0.893 (0.001)	0.888 (0.002)	0.889 (0.002)	0.895 (0.000)
	10	0.625 (0.043)	0.785 (0.032)	0.919 (0.000)	0.918 (0.000)	0.916 (0.000)	0.919 (0.000)
WFG6	3	0.650 (0.020)	0.315 (0.096)	0.676 (0.013)	0.660 (0.012)	0.662 (0.011)	0.685 (0.012)
	8	0.312 (0.140)	0.553 (0.056)	0.870 (0.020)	0.881 (0.019)	0.866 (0.021)	0.869 (0.016)
	10	0.318 (0.154)	0.750 (0.054)	0.888 (0.027)	0.894 (0.018)	0.889 (0.021)	0.886 (0.035)
WFG7	3	0.699 (0.000)	0.202 (0.077)	0.720 (0.003)	0.713 (0.006)	0.710 (0.004)	0.731 (0.000)
	8	0.628 (0.052)	0.560 (0.100)	0.943 (0.001)	0.941 (0.003)	0.938 (0.002)	0.936 (0.037)
	10	0.561 (0.072)	0.713 (0.044)	0.956 (0.026)	0.959 (0.028)	0.960 (0.024)	0.953 (0.031)

Conforme mostrado na Tabela 3, quando se compara o NSGA-III com MOEA/D, este último apresenta piores resultados em termos de Hipervolume para todos menos um caso (WFG1). Ele também possui resultados de qualidade regular em DTLZ4 e DTLZ1 e é incapaz de acompanhar os algoritmos à medida que o espaço de busca aumenta.

Embora os resultados do MOEA/D não sejam estatisticamente superiores na maioria dos testes, ele apresenta uma fronteira de Pareto muito interessante, conforme visto nas Figuras 16(a) e 15(a). Essas figuras mostram que o MOEA/D usando a estratégia de normalização de objetivos pode lidar com objetivos de escala diferentes e que também pode gerar um conjunto distribuído uniformemente de soluções para instâncias de teste com 3 objetivos. No entanto, à medida que o número de objetivos aumenta, fica claro nas Figuras 22(a) e 22(a) que o desempenho de MOEA/D degenera, conforme mostrado pelos baixos valores do hipervolume na Tabela 3.

Semelhante ao MOEA/D, embora os resultados de TwoArch2 não sejam estatisticamente superiores na maioria dos testes, ele também apresenta algumas fronteiras de Pareto interessantes, como pode ser visto nas Figuras 16(b) e 15(b) nas quais as soluções estão principalmente posicionadas próximo ao centro da fronteira de Pareto como pode ser visto na Figura 22(b). À medida que o número de objetivos aumenta, fica claro nas Figuras 22(b) e 21(b) que embora TwoArch2 seja capaz de encontrar soluções em todos os objetivos, ele tem dificuldade em encontrar um conjunto bem distribuído de soluções não dominadas para esta instância, resultando nos pequenos valores de hipervolume mostrados na Tabela 3.

Observando a Tabela 6 nos problemas da família DTLZ (que não há muita distinção de melhores algoritmos, tendo em vista que a diferença do valor do hipervolume nesses problemas é muito baixa, pois a maioria das propostas conseguem valores muito próximos de 1.000) o algoritmo que mais se destacou foi o 2-Greedy, que esteve entre os melhores em 13 das 15 possíveis posições, ficando na frente de algoritmos como NSGA-III, com 11 posições. O resultado disso pode ser visto na fronteira de 2-Greedy nas Figuras 22(j), 21,16 e 15, onde mesmo não conseguindo ter a silhueta mais bem definida e distribuída, conseguiu explorar melhor ótimos locais para garantir uma melhor convergência enquanto possui características similares ao seus pares Convexo e Côncavo.

Para os problemas da família WFG, onde a fronteira de Pareto se torna muito mais complexa devido às propriedades dos problemas dessa família como vista na Seção 5.2.1 é possível observar que o algoritmo que obteve mais sucesso foi o 1-Greedy, com 17 casos de um total de 21, seguido do NSGA-III e R-Greedy com 15 e 14 respectivamente. O algoritmo 1-Greedy por ter menos estágios que o 2-Greedy, consegue se estabilizar mais rapidamente, o que pode explicar a sua melhor performance. É importante ressaltar que o 1-Greedy foi o melhor algoritmo em todos os problemas com 10 objetivos, exceto no WFG1 onde o MOEAD superou todos os outros algoritmos.

Olhando para todos os problemas o algoritmo que esteve mais vezes em primeiro lugar foi o 1-Greedy, seguido pelo NSGA-III, R-Greedy e 2-Greedy. Isso indica que a proposta, dado o tempo natural de convergência de busca pelo melhor conjunto de pontos, pode sim ser competitiva em relação a outros algoritmos do estado da arte.

5.3.4 Análise de complexidade dos algoritmos

O algoritmo K-GreedyMaOEA é uma extensão do NSGA-III. Os principais passos executados no Algoritmo 4 são executados a cada iteração do NSGA-III, após a etapa de normalização. O algoritmo K-GreedyMaOEA é, portanto, um procedimento independente, que modifica os pontos de referência, mantendo todas as demais instruções inalteradas no NSGA-III. O algoritmo é executado a cada Fr iterações. No pior cenário, se $Fr = 1$, as etapas são executadas a cada iteração do algoritmo. As operações mais custosas do loop principal são o loop secundário (etapas 6 ao 12) e o cálculo do hipervolume na etapa 13. O loop secundário é iterado H vezes,

ou seja, é aplicado a todos os pontos de referência. Neste loop, a seleção da transformação matemática (etapa 7) e a seleção do vetor \mathbf{r}_i^t (etapa 8), ambas requerem $O(1)$ computações. A aplicação da transformação matemática no passo 10 requer $O(Km)$, uma vez que transforma cada função objetivo em até K vezes. Assim, como $H \approx N$, o loop secundário requer $O(NKm)$. Assim, como $H \approx N$, o loop secundário requer $O(NKm)$. Agora, com relação ao cálculo do hipervolume, usamos o WFG Hypervolume. A complexidade de tempo do hipervolume WFG, para $m \geq 4$, é $O(N^{m-1})$ (LACOUR; KLAMROTH; FONSECA, 2017). Portanto, a complexidade computacional assintótica do Algoritmo 4 é $O(TN^{m-1})$. A complexidade do pior caso de uma geração de K-Greedy é dominada pelo cálculo do hipervolume e é $O(N^{m-1})$.

5.3.5 Comparação de Tempo de Execução

Esta subseção apresenta uma análise do tempo de execução considerando os métodos propostos e o NSGA-III. O objetivo principal é mostrar que os algoritmos propostos podem ser competitivos e que as mudanças não aumentaram significativamente o tempo de execução. Para realizar essa comparação foi utilizado o teste de Kruskal Wallis, pois como foi discutido, na Seção 5.3.3 se trata de um teste não paramétrico (quando não há certeza a respeito do tipo da distribuição), onde pode-se comparar vários MOEAs de maneira simultânea.

Em relação à análise de tempo, a Tabela 7 apresenta os resultados de tempo de execução para todos os MaOPs utilizados nesse trabalho, considerando 3 números diferentes de funções objetivo. Em cada célula é apresentado o valor médio do tempo de execução para um número específico de funções objetivo e problemas, com o desvio padrão entre parênteses. As soluções que estão em negritos foram aquelas para as quais foi possível rejeitar a hipótese nula (que considera que não há diferença significativa) num nível de significância de pelo menos 5% e que possuem a maior média. Os parâmetros usados para este teste são aqueles usados anteriormente para a criação da Tabela 3. CDTLZ2 e DTLZ2 foram escolhidos devido às suas fronteiras de Pareto inversas.

Tabela 7 – Resultado do tempo de execução (tempo em minutos). Comparação principal Kruskal Wallis

Problema	Objetivos	1-Greedy	2-Greedy	R-Greedy	NSGA-III
CDTLZ2	3	0.564 (0.010)	0.587 (0.022)	0.610 (0.006)	0.064 (0.014)
	8	3.963 (0.961)	4.069 (1.053)	4.111 (1.191)	1.583 (1.104)
	10	147.063 (117.079)	192.511 (156.795)	215.147 (133.357)	171.250 (162.603)
DTLZ2	3	0.645 (0.053)	0.626 (0.034)	0.666 (0.057)	0.064 (0.004)
	8	3.728 (1.282)	3.246 (1.205)	3.377 (1.360)	1.042 (1.240)
	10	65.855 (117.362)	68.866 (106.748)	76.396 (88.996)	63.183 (122.660)

Para o problema DTLZ2, o algoritmo NSGA-III têm os melhores resultados, mas essa diferença diminuí significativamente a medida que o número de objetivos aumenta, chegando a se igualar com 10 objetivos. Um comportamento similar pode ser visto para o problema CDTLZ2, onde o NSGA-III gasta menos tempo que os outros algoritmos, mas à medida que o número de objetivos aumenta essa distância também diminuí até os algoritmos serem estatisticamente iguais.

Apesar de algumas diferenças, todos os algoritmos apresentam resultados de tempo de execução semelhantes, mostrando que as propostas da variante NSGA-III podem ser competitivas no que diz respeito ao tempo de execução a medida que o número de execuções aumenta, assim como pode ser visto na Seção anterior, que diz que a complexidade é dominada pelo o cálculo do hipervolume.

Como esperado, notam-se diferenças maiores no tempo conforme o número de objetivos aumentam. O tempo gasto para 8 objetivos é 6 vezes maior do que o necessário para 3 objetivos e aumenta para 20 vezes quando comparamos 10 com 8 objetivos. Tempos computacionais altos são comuns em algoritmos baseados em indicadores (especialmente aqueles baseados em hipervolume). Assumindo aplicações práticas com $m > 10$, acredita-se que os benefícios do uso de indicadores para orientar a busca podem ser maiores do que as desvantagens.

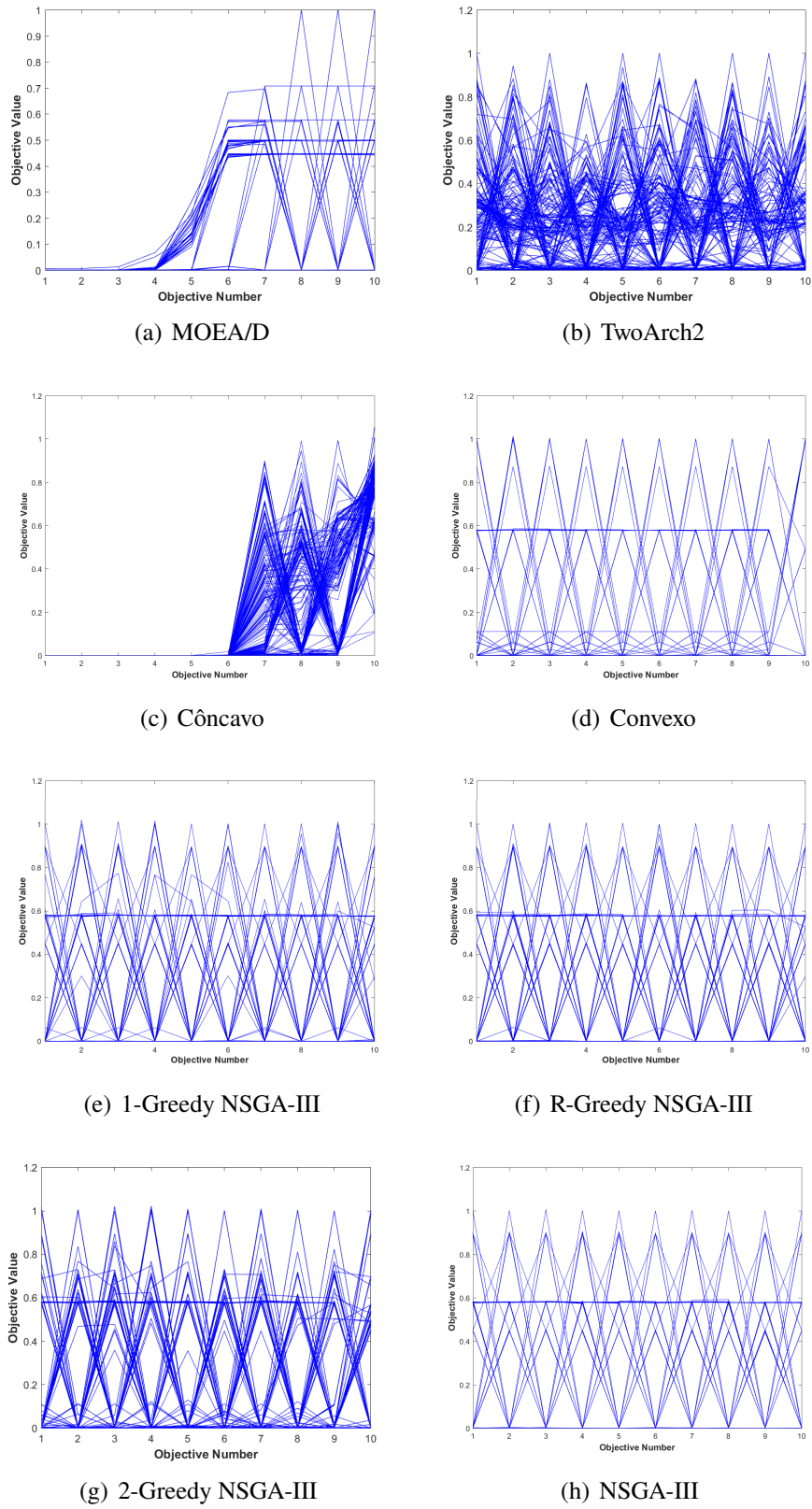


Figura 21 – As coordenadas paralelas da fronteira de soluções não dominadas obtidas por cada algoritmo no DTLZ2 com 10 objetivos na melhor execução associada ao valor do hipervolume normalizado.

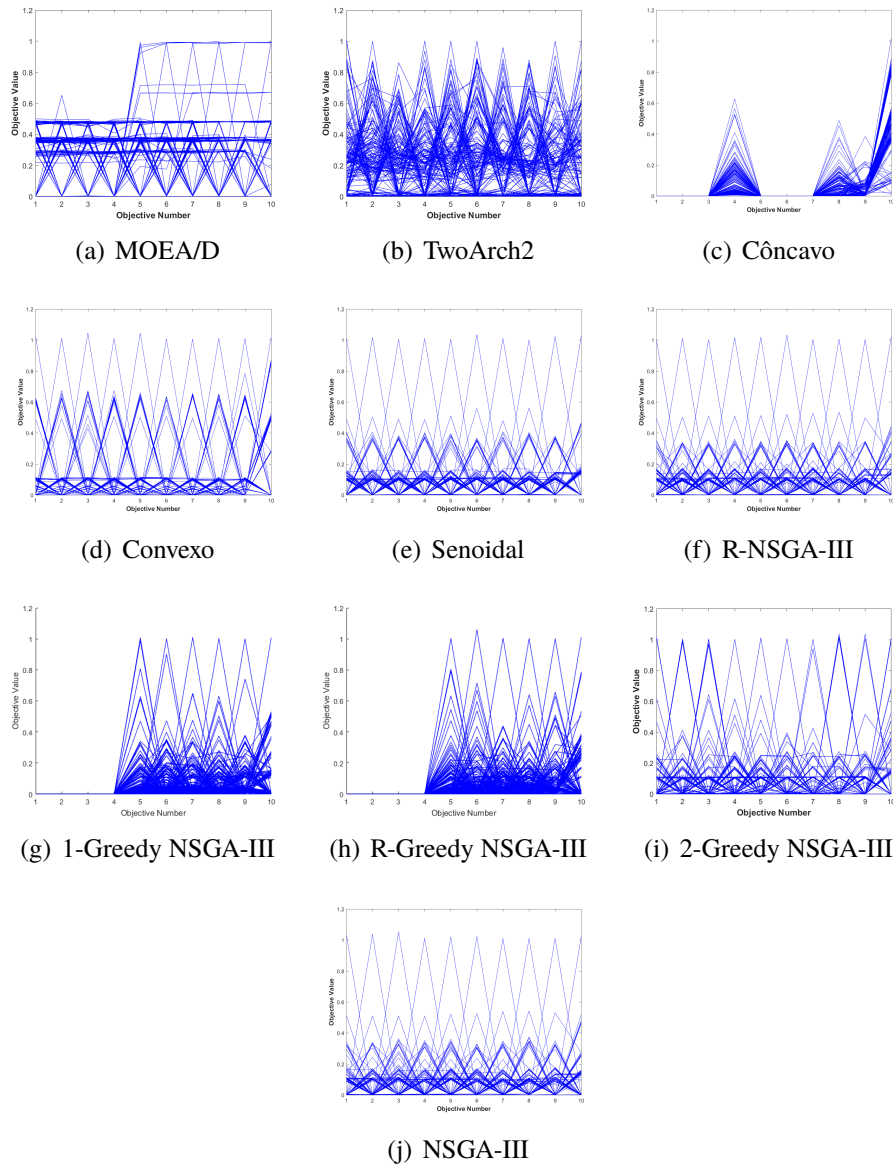


Figura 22 – As coordenadas paralelas da fronteira de soluções não dominadas obtidas por cada algoritmo no CDTLZ2 com 10 objetivos na melhor execução associada ao valor do hipervolume normalizado.

6

Conclusão

6.1 Principais Contribuições

Este trabalho apresentou os resultados de variantes do NSGA-III com o objetivo de investigar a influência da forma das transformações dos pontos de referência no aprimoramento do desempenho da estrutura do NSGA-III na resolução de MaOPs. Três variantes (1-Greedy, 2-Greedy e R-Greedy) incluem definições não determinísticas de que tipo de transformação matemática deve ser considerada e em quais pontos. Enquanto 1-Greedy e 2-Greedy diferem entre si em relação ao valor do parâmetro K (que define o número máximo de vezes em que uma transformação matemática é aplicada em cada ponto de referência selecionado), ambos consideram transformações matemáticas côncava e convexas, enquanto o R-Greedy também inclui a normalização do RVEA no conjunto de possíveis transformações matemáticas. A abordagem K-Greedy representa a principal novidade deste trabalho, pois, até onde é de conhecimento dos autores, esse tipo de transformação nunca havia sido apresentado antes, em especial o R-Greedy, já que neste caso também investigou-se o efeito de usar o procedimento do RVEA (adaptação guiada por vetor de referência), que em algumas iterações específicas transforma o vetor de referência para que cada ponto de referência em cada subespaço possa realizar uma contribuição igual para a população.

6.2 Destaques

Experimentos demonstraram que as transformações nos pontos de referência (principalmente quando côncavas e convexas são usadas como opções intercambiáveis) podem melhorar o desempenho do NSGA-III em alguns casos, sem degenerar os resultados para a maioria dos experimentos quando o número de funções objetivo aumenta. Bons resultados foram alcançados principalmente pelo 2-Greedy, confirmando a hipótese, inicial de que é possível melhorar o desempenho do NSGA-III por meio de transformações de pontos de referência usando formas

diferentes e complementares. As abordagens propostas também lidaram bem com a deterioração da busca quando o número de objetivos aumentou. Quando comparados com outros dois MOEAs (MOEA/D e TwoArch), os resultados mostram que as abordagens propostas também são competitivas com algoritmos de última geração.

6.3 Trabalhos futuros

Para trabalhos futuros, pretende-se explorar alternativas mais sofisticadas (como hiper-heurísticas de métodos não gulosos) para selecionar as transformações matemáticas e pontos de referência, com o objetivo de criar uma hiper-heurística que escolha as melhores opções em um nível mais alto de autonomia e compare os resultados com os MOEAs de última geração. Uma outra alternativa é a utilização de técnicas de aprendizado de máquina, que podem acelerar significativamente o processo de escolha dos melhores pontos de referência, tendo em vista que ela é treinada anteriormente.

Referências

BADER, J. M. *Hypervolume-based search for multiobjective optimization: theory and methods*. [S.l.]: Johannes Bader, 2010. Citado na página 50.

BEUME, N.; NAUJOKS, B.; EMMERICH, M. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, v. 181, n. 3, p. 1653 – 1669, 2007. Citado na página 14.

BRASSARD, G.; BRATLEY, P. *Fundamentals of Algorithmics*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996. ISBN 0-13-335068-1. Citado na página 44.

CHANKONG, V.; HAIMES, Y. *Multiobjective decision making: theory and methodology*. [S.l.]: North Holland, 1983. (North-Holland series in system science and engineering). Citado na página 12.

CHENG, R. et al. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 20, n. 5, p. 773–791, Oct 2016. ISSN 1089-778X. Citado 9 vezes nas páginas 13, 20, 21, 22, 35, 36, 49, 50 e 52.

COELLO, C. A. C.; LAMONT, G. B.; VELDHUIZEN, D. A. V. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387332545. Citado na página 19.

CUATE, O. et al. An approach for the local exploration of discrete many objective optimization problems. *Evolutionary Multi-Criterion Optimization*, v. 10173, p. 135–150, 2017. Citado na página 13.

DAS, I.; DENNIS, J. E. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, v. 8, n. 3, p. 631–657, 1998. Disponível em: <<http://dx.doi.org/10.1137/S1052623496307510>>. Citado na página 20.

DEB, K. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, v. 7, n. 3, p. 205–230, Sept 1999. ISSN 1063-6560. Citado na página 49.

DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *Evolutionary Computation, IEEE Transactions on*, v. 18, n. 4, p. 577–601, Aug 2014. ISSN 1089-778X. Citado 9 vezes nas páginas 13, 20, 22, 25, 32, 33, 36, 44 e 52.

DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, Apr 2002. ISSN 1089-778X. Citado 2 vezes nas páginas 12 e 20.

DEB, K. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, August 2002. Citado na página 26.

DEB, K. et al. Scalable multi-objective optimization test problems. In: *Congress on Evolutionary Computation (CEC 2002)*. [S.l.: s.n.], 2002. p. 825–830. Citado 3 vezes nas páginas 15, 49 e 50.

DEB, K. et al. Scalable Multi-Objective Optimization Test Problems. In: *Congress on Evolutionary Computation (CEC 2002)*. [S.l.]: IEEE Press, 2002. p. 825–830. Citado na página 48.

FLEMING, P. J.; PURSHOUSE, R. C.; LYGOE, R. J. Many-objective optimization: An engineering design perspective. In: COELLO, C. A. C.; AGUIRRE, A. H.; ZITZLER, E. (Ed.). *Evolutionary Multi-Criterion Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 14–32. Citado na página 12.

Gong, D. et al. A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, v. 24, n. 1, p. 142–156, 2020. Citado na página 25.

HUBAND, S. et al. A review of multiobjective test problems and a scalable test problem toolkit. *Evolutionary Computation, IEEE Transactions on*, v. 10, n. 5, p. 477–506, 2006. Citado 6 vezes nas páginas 15, 20, 21, 40, 48 e 49.

HUBAND, S. et al. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, v. 10, n. 5, p. 477–506, Oct 2006. ISSN 1089-778X. Citado na página 49.

Ibrahim, A. et al. Fusion-based hybrid many-objective optimization algorithm. In: *2017 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2017. p. 2372–2381. Citado 3 vezes nas páginas 8, 22 e 23.

Ishibuchi, H.; Doi, K.; Nojima, Y. Reference point specification in moea/d for multi-objective and many-objective problems. In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.: s.n.], 2016. p. 004015–004020. Citado na página 22.

ISHIBUCHI, H.; TSUKAMOTO, N.; NOJIMA, Y. Evolutionary many-objective optimization: A short review. In: *CEC 2008. IEEE Congress on Evolutionary Computation*. [S.l.: s.n.], 2008. p. 2419–2426. Citado 2 vezes nas páginas 13 e 19.

Khan, B. et al. Improved nsga-iii using neighborhood information and scalarization. In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.: s.n.], 2016. p. 003033–003038. Citado 2 vezes nas páginas 22 e 26.

LACOUR, R.; KLAMROTH, K.; FONSECA, C. M. A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research*, v. 79, p. 347 – 360, 2017. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054816301538>>. Citado na página 61.

LI, B. et al. Many-objective evolutionary algorithms: A survey. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 48, n. 1, 2015. Citado 2 vezes nas páginas 13 e 19.

LI, M.; ZHEN, L.; YAO, X. How to read many-objective solution sets in parallel coordinates. *IEEE Computational Intelligence Magazine*, v. 12, 04 2017. Citado 3 vezes nas páginas 36, 39 e 55.

MA, T. et al. Multi-objective optimization intelligent path planning for autonomous driving. *IOP Conference Series: Materials Science and Engineering*, v. 563, p. 052078, 08 2019. Citado na página 12.

- MALTESE, J.; OMBUKI-BERMAN, B.; ENGELBRECHT, A. A scalability study of many-objective optimization algorithms. *IEEE Transactions on Evolutionary Computation*, PP, p. 1–1, 12 2016. Citado na página 19.
- ODU, G. Weighting methods for multi-criteria decision making technique. *Journal of Applied Sciences and Environmental Management*, v. 23, n. 8, p. 1449–1457, 2019. Citado na página 57.
- Oliveira, M.; Britto, A. Influence of reference points on a many-objective optimization algorithm. In: *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.: s.n.], 2018. p. 31–36. Citado 5 vezes nas páginas 14, 16, 25, 37 e 40.
- Oliveira, M.; Delgado, M.; Britto, A. A hybrid greedy indicator- and pareto-based many-objective evolutionary algorithm. *Applied Intelligence*, p. 1–23, 01 2021. Citado 2 vezes nas páginas 14 e 16.
- OLIVER, J. M.; KIPOUROS, T.; SAVILL, A. M. Electrical power grid network optimisation by evolutionary computing. *Procedia Computer Science*, v. 29, p. 1948 – 1958, 2014. ISSN 1877-0509. 2014 International Conference on Computational Science. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050914003561>>. Citado na página 12.
- R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2020. Disponível em: <<https://www.R-project.org>>. Citado 2 vezes nas páginas 36 e 51.
- Riquelme, N.; Von Lücken, C.; Baran, B. Performance metrics in multi-objective optimization. In: *2015 Latin American Computing Conference (CLEI)*. [S.l.: s.n.], 2015. p. 1–11. Citado na página 50.
- SCHÜTZE, O.; LARA, A.; COELLO, C. A. C. On the influence of the number of objectives on the hardness of a multiobjective optimization problem. *IEEE Trans. Evolutionary Computation*, v. 15, n. 4, p. 444–455, 2011. Citado 2 vezes nas páginas 13 e 19.
- TAKAHASHI, R. H. Otimização escalar e vetorial. 2011. Citado 3 vezes nas páginas 12, 17 e 18.
- Tian, Y. et al. An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Transactions on Evolutionary Computation*, v. 22, n. 4, p. 609–622, Aug 2018. ISSN 1089-778X. Citado 3 vezes nas páginas 14, 22 e 31.
- VELASQUEZ, M.; HESTER, P. T. An analysis of multi-criteria decision making methods. *International journal of operations research*, v. 10, n. 2, p. 56–66, 2013. Citado na página 57.
- Vesikar, Y.; Deb, K.; Blank, J. Reference point based nsga-iii for preferred solutions. In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.: s.n.], 2018. p. 1587–1594. Citado 2 vezes nas páginas 22 e 25.
- Wang, H.; Jiao, L.; Yao, X. Twoarch an improved two-archive algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 19, n. 4, p. 524–541, 2015. Citado 2 vezes nas páginas 26 e 52.
- WHILE, R.; BRADSTREET, L.; BARONE, L. A fast way of calculating exact hypervolumes. *IEEE Trans. Evolutionary Computation*, v. 16, p. 86–95, 02 2012. Citado na página 50.

- ZHANG, Q.; LI, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, v. 11, n. 6, p. 712–731, dec. 2007. ISSN 1089-778X. Citado 2 vezes nas páginas 23 e 26.
- ZHANG, Q.; LIU, W.; LI, H. The performance of a new version of moea/d on cec09 unconstrained mop test instances. In: *Evolutionary Computation, 2009. CEC '09. IEEE Congress on.* [S.l.: s.n.], 2009. p. 203–208. Citado na página 26.
- ZHANG, Y. et al. Binary differential evolution with self-learning for multi-objective feature selection. *Information Sciences*, v. 507, 08 2019. Citado na página 26.
- ZHANG, Y. et al. A decomposition-based archiving approach for multi-objective evolutionary optimization. *Information Sciences*, v. 430, 11 2017. Citado na página 25.
- Zhao, M. et al. A many-objective evolutionary algorithm with fast clustering and reference point redistribution. In: *2018 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2018. p. 1–6. Citado 4 vezes nas páginas 8, 22, 24 e 25.
- ZHOU, A. et al. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, v. 1, p. 32–49, 03 2011. Citado na página 19.
- ZITZLER, E.; KUNZLI, S. Indicator-based selection in multiobjective search. In: *in Proc. 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*. [S.l.]: Springer, 2004. p. 832–842. Citado na página 14.